Information Sciences 625 (2023) 342–366

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Counterfactual explanation generation with minimal feature boundary

Dianlong You [a,b], Shina Niu [a,b], Siqi Dong [a,b], Huigui Yan [a,b], Zhen Chen [a,b], Di Wu [c,*], Limin Shen [a,b], Xindong Wu [d]

[a] School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China
[b] The Key Laboratory for Software Engineering of Hebei Province, Yanshan University, Qinhuangdao, Hebei 066004, China
[c] College of Computer and Information Science, Southwest University, Chongqing 400715, China
[d] The Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Article history:
Received 13 September 2022
Received in revised form 16 December 2022
Accepted 1 January 2023
Available online 5 January 2023

Keywords:
Counterfactual explanation
Feature boundary
Interpretability
Causal discovery

ABSTRACT

The complex and opaque decision-making process of machine learning models restrains the interpretability of predictions and makes them cannot mine results outside of learning experiences. The causality between features and the target variable can be traced by injecting counterfactual explanations into the prediction model and generating counterfactual instances using adjusted features to reverse the prediction results. Existing algorithms, such as Diverse Counterfactual Explanations (DiCE) and Counterfactual Explanations Guided by Prototypes (Proto), can generate multiple/single counterfactual(s) for a data point by global optimization in the range of all-out features to gain on a local decision range. However, these methods cannot clearly identify which features are the key causes. Moreover, a Random Forest Optimal Counterfactual Set Extractor (RF-OCSE) extracts counterfactual sets from a random forest and needs to manipulate all the internal nodes of the tree, restricting it to only tree ensemble models. To address the above shortcomings, we proposed a Counterfactual Explanation Generation method with the Minimal Feature Boundary (MFB), named (CEG_{MFB}). The proposed CEG_{MFB} algorithm consists of two stages: 1) mining the MFB, which can reverse the prediction results to restrain the generation range of counterfactual instances, and 2) constructing a counterfactual generative method for generating counterfactual instances within the MFB to realize the minimum reversing cost. To evaluate its performance, we compared the proposed CEG_{MFB} algorithm with six baseline algorithms on 16 datasets and conducted a case study in a real scenario. The results indicate that the proposed CEG_{MFB} algorithm outperforms the compared algorithms.

1. Introduction

Exploring the causal mechanism behind massive, high-dimensional, and observational data can improve the interpretability of model prediction, which is a current research hotspot [1–3]. In machine learning (ML), systems such as deep neural networks are black-box in nature [3], which means that although accurate prediction results can be obtained, causality cannot be explained [4]. In addition, it is difficult for humans to understand due to the complex and opaque

* Corresponding author.
E-mail addresses: youdianlong@sina.com (D. You), wudi.cigit@gmail.com (D. Wu).

https://doi.org/10.1016/j.ins.2023.01.012
0020-0255/

decision-making process. Interpretable ML has become a strong competitor for black-box models, especially when it is used to provide decision-making information in key areas, such as finance, health care, education, and criminal justice [5]. For example, if a loan application is rejected by the bank, the reason may be a "bad deposit record." However, such an explanation does not help the lender decide what to do next. The rejected lender usually thinks about a question: what changes does the bank need to make to approve my loan? Well-reasoned, counterfactual explanation can provide further information, such as "if your income is higher than $5000, you will get a loan." In addition to the modification of features to achieve the desired results, counterfactual explanation can also find fairness problems [6,7], such as gender and racial discrimination. Specifically, once the counterfactual instances obtained always modify the features in original instances that cannot be changed, such as race, it indicates that racial discrimination exists in reality as only when race changes can we achieve the desired results.

Counterfactual explanation makes model decisions by generating counterfactual instances/examples from the original ones [8,9], belonging to "inferring the cause from the result", which negates and restates the past facts [10,11]. Generally, mining the most influential features can contribute to a model's interpretability for a specific decision by imitating an opaque model, such as SHapley Additive exPlanations (SHAP) [12] and Quantitative Input Influence (QII) [13]. However, counterfactual instances are not generated, resulting in a lack of deeper understanding of explanations and specific recommendations. Furthermore, counterfactual is the top level of causality and allows causal explanations to be ascribed to data. However, existing methods focus only on prediction reversion using the objective function and fail to identify causality, which may lead to the interpretation of spurious causality and significant additional costs. Specifically, the unsolved defects can be summarized as follows: 1) Some methods, such as genetic algorithm, random forest, and KD-tree, can construct counterfactual instances by optimization. However, the generated instances contain many irrelevant and redundant features, resulting in the instability of decisions; 2) Gradient-based counterfactual algorithms, such as Diverse Counterfactual Explanations (DiCE) [14], adds a diversity measure to the objective function to generate multiple counterfactual instances for a single original instance; Counterfactual Explanations Guided by Prototypes (Proto) [15] uses prototypes to help generate counterfactual instances; Model-approx CF [16] captures the monotonic trend to generate actionable counterfactual instances by modifying the variational autoencoder loss. However, these methods cannot accurately identify causal features for the decision, which results in many useless changes; 3) Some methods are constrained to specific models. For example, because Random Forest Optimal Counterfactual Set Extractor (RF-OCSE) needs to access the interior of the model to generate counterfactual instances, it can only handle tree ensemble models [17].

Therefore, to solve the unsolved defects in this area, identifying the causality in a dataset, limiting the generation of counterfactual instances to the range of the causality, and mining the feature boundary to exclude many irrelevant and redundant features are reliable solutions. In this regard, we propose a Counterfactual Explanation Generation method with Minimal Feature Boundary (MFB), named (CEG_{MFB}). The challenges of implementing CEG_{MFB} lie in two aspects: 1) how to mine the boundary of feature changes to constraint minimal features with high causality to generate counterfactual instances, and 2) how to revise the minimal features with a minimum cost to obtain the prediction results of counterfactuals. In Fig. 1, the yellow area indicates a valid counterfactual instance space. The red rectangle indicates the MFB. The features in MFB maintain highly casual relationships with the target variable. The $x^{cf1}, x^{cf2}, x^{cf3}, x^{cf4}, x^{cf5}$, and x^{cf6} are six types of possible counterfactual instances generated by instance x. Among them, the x^{cf1}, x^{cf2}, and x^{cf3} are generated under the MFB, and x^{cf4}, x^{cf5}, and x^{cf6} are generated using all features. Meanwhile, x^{cf1} and x^{cf4} are invalid counterfactual instances because they do not realize the reversal of the prediction results. In contrast, the $x^{cf1}, x^{cf3}, x^{cf4}$, and x^{cf6} are valid. In addition, because x^{cf5} and x^{cf6} are generated using all features, their causal interpretability is unconvincing. Although x^{cf2} and x^{cf3} generate counterfactual instances in MFB, x^{cf2} is more outstanding than x^{cf3} due to its closer distance to x. Note that x^{cf2} is generated under the MFB with the closest distance x, and we call x^{cf2} the optimal counterfactual instance of x. Therefore, the goal of the counterfactual explanation method CEG_{MFB} is to generate counterfactual instances like x^{cf2}.

To this end, we propose a novel CEG_{MFB} algorithm based on two main ideas: 1) Considering that causal features in mined Markov blanket (Mb)/Markov Boundary (MB) are not unique/incomplete when the faithfulness condition is not satisfied (see Section 4.1.2), we take advantage of the fact that the mutual information gains of features in the MFB increase with the addition of new causal features to further mine causal features from overall features. In detail, we introduce the add-factor parameter θ and automatically search for the optimal θ by iteratively calculating the validity of counterfactual instances and gradually adding new causal features to MFB until the termination conditions are satisfied to obtain the final MFB; 2) Considering causal invariance, we construct an optimization strategy to generate the expected counterfactual instances within the MFB and realize the minimum reversing cost by minimizing the distance between the original and the counterfactual instances.

The main contributions of this paper are summarized as follows:

- We initially introduce a term called MFB and construct its mining algorithm minFB to constrain the generation range of the counterfactual explanation. The features in MFB maintain high causality with the target variable, which can improve the interpretability of classification models. Meanwhile, the complexity of features can be significantly reduced.
- We originally propose a novel method CEG_{MFB} to generate counterfactual instances using causal features in MFB instead of all features. CEG_{MFB} can achieve counterfactual results by modifying partly selected features within the MFB to reverse classifier predictions.

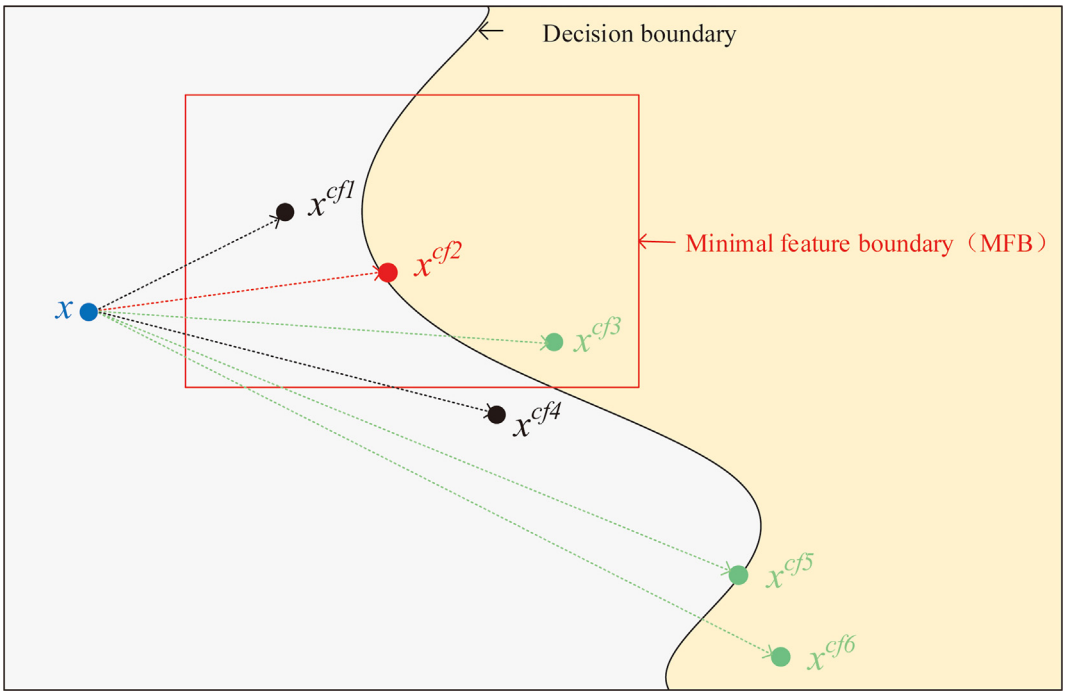


Fig. 1. A counterfactual instance space. The instance x and its possible counterfactual instances $x^{cf1}, x^{cf2}, x^{cf3}, x^{cf4}, x^{cf5}$, and x^{cf6}. The $x^{cf1}, x^{cf3}, x^{cf4}$, and x^{cf6} are valid, while x^{cf1}, x^{cf4} are invalid.

- We analyze the proposed CEG_{MFB} algorithm and evaluate its performance on 16 datasets and a real scenario in terms of relevant evaluation metrics, including validity, proximity, sparsity, and distance, and compare it with six representative counterfactual explanation generation algorithms. The experimental results indicate that CEG_{MFB} outperforms the compared algorithms, especially on high-dimensional data.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 introduces the problem settings. In Section 4, we construct minFB algorithm to mine MFB, then propose our counterfactual explanation generation algorithm CEG_{MFB}. Section 5 presents the experimental studies, which is divided into general settings, experimental results, and a case study on a real scenario. Finally, Section 6 concludes the paper.

2. Related Work

Machine learning with a causal mechanism is a current research hotspot with five major research areas, i.e., causal supervised learning, causal generative modeling, causal explanations, causal fairness, and causal reinforcement learning; the causal explanation technique is divided into feature attribution and contrastive explanations [18,14]. Attribution-based explanations assign ranks to features, representing the contribution to the output of the model. Contrastive explanations are typically counterfactual by estimating an altered version of an observation to change the model's prediction [8]. Counterfactual explanation explain a prediction by computing the change in the original features to classify it in the desired class. The open challenges in causal explanations include unifying feature attribution and explanations, scalability and throughput, dynamics, security, privacy, robustness, sensitivity, etc. [18] In this study, we focus on feature attribution and counterfactual explanation.

2.1. Interpretability of Machine Learning

The interpretability problem in ML can be divided into model and result explanations [19]. The model explanation uses inherently interpretable and transparent models to search for a global explanation of the original model, including the use of linear and logistic regressions, decision trees, and rule sets [20]. Since many ML algorithms are black-box to the end user and do not guarantee input–output relationships, to improve the interpretability and credibility of algorithmic decisions and enhance human decision making, outcome explanation generates post hoc explanations for opaque models [21–23]; such explanations need to provide explanations for specific predictions in the model and need not be global or explain the internal logic of the model.

Interpretability research focuses on providing explanations for specific predictions in one model [8]. In addition to feature attribution methods, such as LIME [24], SHAP [12], and QII [13], counterfactual explanation method which generates counterfactual instances is another interpretability technique used to explain specific results, and several important studies have been conducted on it. Because LIME, SHAP, and QII, may generate inaccurate descriptions of how the input data affect the system decision-making, the counterfactual explanation is better than the feature attribution method [25].

2.2. Counterfactual Explanation

Counterfactual, which is the highest level of the causality ladder [26], will trigger human causal reasoning [27] and can verify the concept of causal thinking through itself. As an example-based method, counterfactual explanation describes a situation in this form: "if X did not happen, Y would not happen." Therefore, counterfactual explanation can be used to explain systematic behavior in the narrow sense of causality.

Wachter et al. [9] first proposed counterfactual explanation and generated counterfactual instances by optimizing the objective function. Since then, most counterfactual generation techniques have relied on establishing an optimization problem to find the nearest counterfactual instance related to the original instance to be interpreted in the feature space [28]. Generally, counterfactual explanation generation algorithms can be divided into three categories according to different access levels of the basic model of the generated counterfactual.

- **Access to complete model internals.** When the algorithm uses solver-based methods, such as mixed-integer programming [29–31], or the algorithm runs on the decision tree [17], it needs to access the complete interior of the model. Therefore, a major of these algorithms are restricted to a specific model. For example, RF-OCSE is based on a partial fusion of tree predictors from an RF into a single Decision Tree (DT) and obtains a counterfactual set. However, RF-OCSE needs to access all internal nodes of the tree and only work for the tree ensemble models. Russell et al. [32] proposed a novel set of constraints and used it with an integer programming solver to find coherent counterfactual explanations, which required the linear model only.
- **Access to gradients.** This is the most common technique in counterfactual explanation generation algorithms. Gradient optimization is used to solve the optimization objective by modifying the objective function proposed by Wachter [9]. Model-approx CF captures the monotonic trend by learning the linear model between two variables. It presents a simple approximation loss to replace the proximity term to achieve feasibility and uses a variational autoencoder as the CF generator. However, Model-approx CF governed by domain knowledge only approximates causality by monotonicity. DiCE incorporates diversity into the optimization policy to generate a diverse set of counterfactual explanations based on a determinantal point. Proto uses class prototypes to mine counterfactual explanations [33]. Class prototypes can be obtained using an encoder or KD-trees. However, they cannot identify which features are the key causes. Meanwhile, irrelevant and redundant features can result in shaming the decisions of the classifier.
- **Access to only the prediction function.** The black-box method, without accessing any model interior and gradient, uses the gradient-free optimization method, such as Nelder-Mead [34], growing spheres [35], FISTA [36], or genetic [37] algorithm to solve the optimization problem. The DiCE-Random, DiCE-Genetic, and DiCE-KDTree, as the black-box versions of DiCE and the representative of the black-box algorithm, solve the optimization problem by using random, genetic, and KD-tree algorithms, respectively. They only access the prediction function to reverse the prediction for generating counterfactual explanation. The model is immaterial for these black-box methods, such that the model types are not constrained. Unfortunately, due to the lack of accurate feature boundary, the classifier will produce abundant sham judgment once the counterfactual instances are generated by global optimization and reduce the validity of counterfactual explanation.

For the aforementioned counterfactual explanation methods, 1) the dimensionality of the feature space or the number of sampled observations increases proportionally during the counterfactual explanation search, leading to the sham judgment of the classifier and computational bottlenecks [15]; 2) these existing methods attempt to find modifications leading to desired results through global optimization, such that the underlying causality from features is unreliable. Therefore, integrating causality into the generation of counterfactual instances and reducing the reverse scale of features by discarding irrelevant and redundant features are imperative issues for counterfactual explanation. In these contexts, we attempt to generate counterfactual instances by revising the causal features in the original instances, which can clearly identify the cause and reduce the false decision. In addition, by limiting the feature range to causal features, we can reduce the impact of high-dimensional data. In contrast, our CEG_{MFB} algorithm does not perform global optimization. It can find the potential causality while reducing the range of features by obtaining MFB and generating counterfactual instances within MFB by optimizing minimum costs with the causal invariance.

3. Problem Settings

In this section, we describe the related problem settings, notations, and definitions. Table 1 summarizes the notations used in this paper and their meanings. Under the constraint of independent identically distributed (IID), we can mine the

Table 1
Summary of Notations and Meanings.

Notations	Meanings
F	d-dimensional feature/ variable set, $F = \{F_1, F_2, \ldots, F_d\}$
F_i	the *i-th* feature/ variable
f	value of a feature/ variable
T	the target variable
t	value of a target variable
V	a feature/ variable set, $V = F \cup T = \{v_1, v_2, \ldots, v_{d+1}\}$
x	an original instance
x^{cf}	a counterfactual instance of x
D	a dataset
MFB	the Minimal Feature Boundary
Mb, Mb_T	Markov blanket; Markov blanket of T
MB, MB_T	Markov Boundary; Markov Boundary of T
PC, PC_T	Parents and Children; Parents and Children of T
SP, SP_T	Spouses; Spouses of T
SF	the selected feature set
DF	the abandoned feature set
θ	the add-factor
$do(\cdot)$	the intervention operator
$Ind(F_i, T\|S)$	F_i and T are conditionally independent given $S, S \subseteq F \setminus F_i$, i.e., $F_i \perp\!\!\!\perp T\|S$
$Dep(F_i, T\|S)$	F_i and T are conditionally dependent given $S, S \subseteq F \setminus F_i$, i.e., $F_i \not\perp\!\!\!\perp T\|S$

unique MFB of the target variable. We only take binary classification tasks into account because multi-class classification problems can be divided into several binary classification subtasks utilizing One-vs-Rest or One-vs-One techniques.

Definition 1 (*Bayesian Network, BN [38]*). A Bayesian network is a directed acyclic graph, which can be represented as a triple (V, G, P(V)). G is a set of directed edges, $G = \{< v_i, v_j > | v_i \neq v_j \text{ and } v_i, v_j \in V\}$, each edge $< v_i, v_j >$ represents the variables v_i, v_j have causal relationship (v_i is the cause of v_j, v_j is the effect of v_i). P(V) is a set of conditional probabilities that can be decomposed as a product of conditional probabilities as follows:

$$P(V) = \prod_{v_i \in V} P(v_i | Pa(v_i)) \tag{1}$$

in which $Pa(v_i)$ denoted the parents of v_i.

Definition 2 (*Faithfulness [38]*). Given a Bayesian network $< V, G, P(V) >, G$ is faithful to $P(V)$ if and only if every conditional independence present in P is entailed by G and the Markov condition. $P(V)$ is faithful if and only if G is faithful to $P(V)$.

Definition 3 (*Markov blanket,Mb [38]*). A set of variables is a Markov blanket of the target variable T (Mb_T) if and only if for $\forall F_i \in F \setminus Mb_T, T \perp\!\!\!\perp F_i | Mb_T$.

The goal of counterfactual explanation is to generate counterfactual instances [8] by finding the minimum disturbance of the original instances and counterfactual ones while maximizing the differences between the prediction results and real ones. The counterfactual instance x^{cf} of x holds two properties: 1) The prediction result of x^{cf} on the classifier is opposite of the original x; 2) The changes from x to x^{cf} should be as minimal as possible. For the generation of counterfactual instances, we iteratively interfere with the features in MFB by optimization function.

The Challenges of Counterfactual Explanation Generation:

Challenge I. *Difficulty of avoiding false features when explaining the causality between features and the target variable.* Because datasets contain many irrelevant or redundant features, classifiers' false judgments will have an impact on their counterfactual ability. Exploring the cause and effect before the counterfactual explanation and mining the appropriate feature range are crucial solutions.

Challenge II. *Difficulty in generating counterfactual instances with a minimum cost of reversing prediction results.* Among the many counterfactual instances generated by an instance, we frequently pursue the instance that can bring counterfactual prediction results at the minimum cost. The challenge is determining how to achieve the minimum change in the selected features through the optimization function to simply seek the decision boundary of the counterfactual instances.

Above all, we refer to this mining range as the minimal feature boundary (MFB) and design the algorithm for the mining range, named minFB. Then, based on minFB, a counterfactual explanation generation algorithm with the minimal feature boundary is proposed, named CEG_{MFB}.

4. Counterfactual Explanation Algorithm with Minimal Feature Boundary

4.1. Mine the Minimal Feature Boundary (MFB)

In this section, we present how to mine the MFB, which comprises causal features for generating counterfactual instances. First, we attempt to find most of the causal features by *MB*. Then, considering that the *Mb* is not unique when the faithfulness assumption is not satisfied and the mined causal features in the *MB* may be incomplete, we introduce the add-factor θ to supplement the remaining causal features to MFB by calculating iteratively the validity of counterfactual instances.

4.1.1. Mine Causal Features Based on MB

A large number of irrelevant and redundant features can cause sham judgments of the classifiers, which cannot accurately judge the main reasons for the change of the target variable. Once a classifier generates false judgments, the desired results by the optimization function will force the false features to change, which creates challenges in generating a valid counterfactual instance. Therefore, we propose using the features in the MFB to forecast the target variable T to avoid some misleading feature changes. Initially, we explore the causality between features and the target variable T using *MB* to discover the majority features in MFB.

Definition 4 (*Markov Boundary, MB [38,39]*). If Mb_T does not have a proper subset satisfying the definition of *Mb*, then the Mb_T is called the Markov Boundary of T, named MB_T. In BN, the MB_T consists of its direct causes, direct effects, and other direct causes of its direct effects (i.e. its PC_T and SP_T variables) under faithfulness assumption.

Definition 5 (*Conditional Independence, CI [39]*). A feature F_i and T are said to be conditionally independent given S (denoted as $Ind(F_i, T|S)$, if and only if $P(F_i, T|S) = P(F_i|S)P(T|S)$. Otherwise, F_i and T are conditionally dependent given S, denoted as $Dep(F_i, T|S)$.

Proposition 1 [40]. Given two features F_i and $F_j, F_i \in PC_T$ if and only if $Dep(F_i, T|S)$ for all $S \subseteq F \setminus F_i$. And assuming that $T \rightarrow F_i \leftarrow F_j$ or $F_j \rightarrow F_i \leftarrow T, F_j \in SP_T$ if $\exists S \subseteq F \setminus F_i, F_j$ such that $Ind(T, F_j|S)$ and $Dep(T, F_j|S, F_i)$.

Under the faithfulness assumption, there is only one Mb_T, which is the MB_T according to Definition 4. Therefore, we only need to focus on MB_T, which contains all the information to predict T under faithfulness assumption because for $\forall F_i \in F \setminus MB_T, T \perp\!\!\!\perp F_i|MB_T$. In addition, according to the independence attribute satisfied by the features in the MB_T, we usually use the CI test to find the MB_T, in which the PC_T and SP_T are found according to Proposition 1. However, when the faithfulness assumption is not satisfied, there are multiple Mb_T, and the *MB* algorithm may only mine part of the dependencies. At this moment, the MB_T cannot mine all causal features to accurately predict T, so it cannot be used for optimization to generate valid counterfactual instances. In order to deal with this situation, we give the concept of MFB.

Definition 6 (*Minimal Feature Boundary, MFB*). The minimal feature boundary is a minimal feature set that approximates the Markov Boundary and can cover the high causality features of T for generating counterfactual instances when not satisfying the faithfulness assumption.

Proposition 2. The features in mined MB_T by MB algorithms contain partial (all) causal features in MFB.

Proof 1. Based on Definitions 4 and 5, we can obtain that $P(T|MB_T, S) = P(T, MB_T)$ for $\forall S \subseteq F \setminus MB_T$, which indicates that features contains in MB_T has causality with T. Moreover, according to Definition 4, the Mb_T may not be unique when not satisfying the faithfulness assumption. And if there is more than one Mb_T, obtained MB_T only reflects a portion of dependencies in the data. Furthermore, the *MB* methods may miss important features due to the influence of noise and PCMasking [41] caused by incorrect conditional independence (CI) tests. The PCMasking phenomenon is illustrated in Fig. 2. PCMasking may cause some PC_T to be discarded (i.e., MaskingPC features) as well as SP_T selection problems due to an incorrect CI test. Therefore, the causal features of T mined by *MB* may be missed. The proof of Proposition 2 is complete.

According to Proposition 2, we can mine partial (all) causal features by *MB*, and then, in order to supplement the missing causal features and mine complete MFB, we introduce add-factor θ in the next section.

4.1.2. Mine Remaining Causal Features by Using the add-factor θ

Due to mined features being incomplete using *MB* according to Proposition 2, we add missed features to the MFB after MB_T to ensure the validity of the generated counterfactual instances, despite the idea of using MB_T as the minimal feature boundary is intuitive. We introduce the add-factor θ for supplementing MFB by using mutual information. Mutual information is based on the entropy of variables. Given a variable F_i, the entropy of F_i is defined as:

Fig. 2. PCMasking phenomenon. The white nodes are included in PC_T first, and then, the gray nodes, i.e., MaskingPC features, are discarded due to incorrect CI. (a) If $Ind(T, F_1|\{F_2, F_3\})$, MaskingPC is $\{F_1\}$. (b) If $Ind(T, \{F_2, F_3\}|F_1)$, MaskingPC is $\{F_2, F_3\}$.

$$H(F_i) = -\sum_f P(f) log P(f) \tag{2}$$

The entropy of F_i after observing values of another variable T is defined as:

$$H(F_i|T) = -\sum_t P(t) \sum_f P(f|t) log P(f|t) \tag{3}$$

Definition 7 (*Mutual Information, MI [42]*). Mutual information (MI) is a useful measure of information that can be though as the amount of information contained in a random variable about another variable, or the reduction of uncertainty in a random variable due to the knowledge of another random variable.

In Eq. (2) and Eq. (3), $P(f)$ is the prior probability of $F_i = f$, and $P(f|t)$ is the posterior probability of $F_i = f$ given $T = t$. From Eqs. (2) and (3), the value of the mutual information between F_i and T, denoted by $I(F_i; T)$, is defined as:

$$I(F_i; T) = I(T; F_i) = H(F_i) - H(F_i|T) = \sum_{f \in F_i} \sum_{t \in T} P(f, t) log \frac{P(f, t)}{P(f)P(t)} \tag{4}$$

Mutual information can measure the dependence degree between F_i and T. Meanwhile, the features that have causality on T obviously have high MI values. Thus, we use mutual information as the metric and introduce the add-factor θ which represents the number of expanding features having high MI values. We calculate the MI values between the features not selected by MB and T, then rank the features according to the values, and finally add ranked features under the control of add-factor θ to the MFB.

Since the number of features in the MFB of each dataset is different, we set the choice of the add-factor θ as an iterative process. A counterfactual instance is invalid when it makes a change to the original instance but does not successfully reverse the prediction of the T. Mining MFB aims to find the valid generating range of counterfactual instances; thus, the validity can be the condition to determine whether the mining process should be stopped as an evaluation metric. Therefore, we define a validity evaluation function $evaluate\_validity$ that measures the validity of generating counterfactual instances within SF using the generation mechanism in Section 4.2 and take the validity as the condition for the stop of add-factor θ iteration. The add-factor θ starts from zero, increases in turn, and calculates the validity. After the iteration process, the expanded features controlled by add-factor θ are combined with the selected MB_T features, which is the final MFB. Fig. 3 presents an MFB of T, where we first mine the most causal features of T by using MB, i.e., the blue features, and then mine the remaining causal features by add-factor θ in the case of missing features, i.e., the green extended features. We optimize within MFB to find the minimum feature change reversing T.

4.1.3. minFB Algorithm

The proposed minFB algorithm is presented in Algorithm 1, which includes two stages. Stage 1 is to mine the causal features of the target variable T by MB at lines 2–9. The two steps include mining the MB_T at lines 3–4 and calculating the mutual information values between each feature in the set of $F \setminus MB_T$ and T at lines 5–9. Stage 2 is to mine the remaining causal features from MF (sorted IF) using the add-factor θ. The two steps include sorting features in $F \setminus MB_T$ by descending order according to the mutual information value at line 8 and mining the remaining features using the iteration of the add-factor θ, which are lines 13–19.

For the minFB algorithm, after identifying MB_T, we iteratively mine the remaining causal features from MF. We use the validity of the counterfactual instances generated within SF as the judgment and termination conditions of θ iterations. Once the validity==100% or the validity no longer increases, the iterative process of mining the remaining features will stop. This means that the remaining features in MF cannot increase the validity. That is, SF contains the maximum validity of the causal

features required for optimization. Considering that the features in *MF* are to participate in the iteration of the add-factor θ in descending order of mutual information with the target variable T at lines 10–12, the top-ranked features are more likely to be added to *SF*. If the consecutive remaining features cannot increase the validity, the validity will more likely not be increased by using the subsequent features in *MF*. Therefore, we set up three iterations whose validity is no longer increasing that represent "validity stops increasing" to reduce the computational complexity and errors in subsequent optimizations for counterfactual instance generation. For the experimental results on the effect of the iteration numbers, we will demonstrate their performance in Tables 5, 6 in "Section 2.2.1(2) Effectiveness of the add-factor θ." The experiments demonstrate that validity is not increased by adding more iterations if it does not increase after three consecutive iterations. Under this constraint, the generated *SF* can ensure that the counterfactual instances have the highest validity while making fewer changes to the original instances.

For the *evaluation_validity* function, we take D, F, and T as the basic parameters, input *SF* obtained in this iteration. Then, we generate counterfactual instances by modifying the features in *SF* based on the generation mechanism in Section 4.2. Meanwhile, validity is calculated using Eq.(16) to determine whether *SF* in this iteration has met the requirements. We employ the HITON-MB (HITON-PC and its find Spouse algorithm) algorithm in steps 3–4 to mine MB_T, one of the best *MB* discovery algorithms [43]. We can use any other up-to-date *MB* algorithms to replace HITON-MB.

Algorithm 1: The minFB Algorithm

Input:The dataset D; The feature set F; The target variable T; The add-factor θ;
Output: The selected feature set as the result of MFB: *SF*;
1. $MB_T = \phi; SP_T = \phi; MI(F) = \phi; \theta = 0$
2. /*stage 1: mine causal features by *MB* */
3. find PC_T and SP_T in dataset D according Proposition 1
4. $MB_T = PC_T \cup SP_T$;
5. $IF = F \setminus MB_T$
6. **for** F_i in *IF* **do**
7. /* compute mutual information of feature F_i by Eq.(4) */
8. $MI(i) = I(F_i; T)$
9. **end for**
10. /*stage 2: mine remaining causal features using add-factor θ */
11. /* sort features based on $MI(i)$ */
12. $MF = sort(IF, MI)$
13. **repeat**
14. /* the iteration of add-factor θ */
15. $SMF = select(MF, \theta)$;
16. $SF = MB_T \cup SMF$;
17. $validity = evaluate\_validity(SF, D, F, T)$;
18. $\theta = \theta + 1$;
19. **until** validity == 100% or validity stops increasing;
20. output *SF*

4.2. Counterfactual Explanation by Generating Counterfactual Instances

4.2.1. Preliminaries

In this section, we further analyze the necessity of mining MFB from the perspective of counterfactual explanation through two problems.

Problem 1: *Why choose SF to generate counterfactual instances?*

The existing representation learning methods can maximize the prediction accuracy on the verification set; however, they cannot prevent false features, which is unfavorable for the generation of counterfactual instances to interpret prediction. For example, it produces a "grass" false feature when it recognizes "dog", such that the counterfactual instance will alter the "grass" feature to reverse the "dog" prediction. However, "grass" is obviously not the true reason for being classified as a dog. To improve the interpretability of classification results, we define a generation range of counterfactual instances that should be included in the learning objectives.

Proposition 3 [44]. Take a feature $F_i \in F, F_i = f$ as the potential cause of the target variable T, $T = t$, then a non-spurious feature should be a sufficient cause of the prediction and is measured by probability of sufficiency (PS) and the formula is as follows:

$$PS_{F_i=f,T=t} = P(T(F_i = f) = t | F_i \neq f, T \neq t) \tag{5}$$

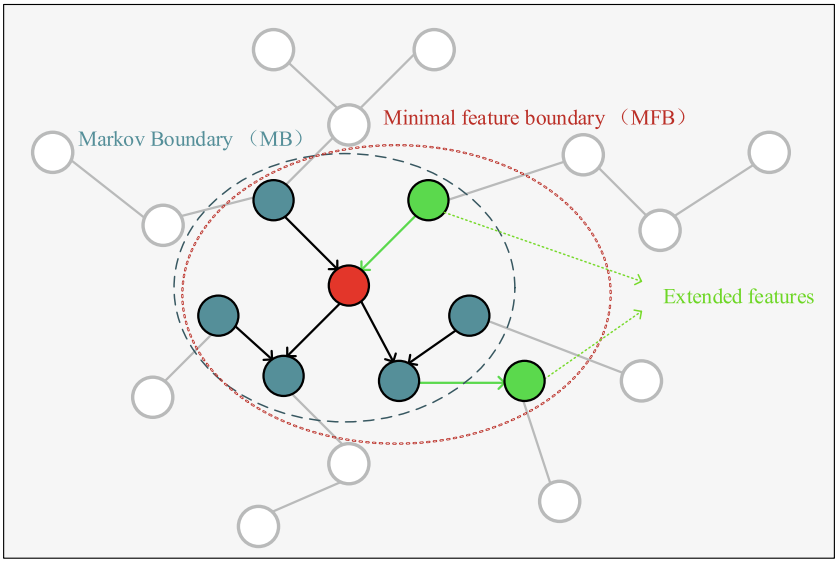


Fig. 3. The minimal feature boundary of target variable T. The red, blue and green nodes indicate T, causal features in MB_T, expanded causal features, respectively. The MFB is expanded from the blue nodes to the green ones.

Proposition 4. [44]Take a feature $F_i \in F, F_i = f$ as the potential cause of the target variable T, $T = t$, then an efficient representation must capture features that are necessary causes of the prediction. Quantify efficiency is measured by probability of necessity (PN) and the formula is as follows:

$$PN_{F_i=f,T=t} = P(T(F_i \neq f) \neq t | F_i = f, T = t) \tag{6}$$

Probability of necessity and sufficiency (PNS) are used to quantify non-spurious and efficiency simultaneously. The formula is as follows:

$$PNS_{F_i=f,T=t} = P(T(F_i \neq f) \neq t, T(F_i = f) = t)) \tag{7}$$

Proposition 5. The minimal feature set (SF) we mined should be included in the learning objective for counterfactual purpose.

Proof 2. For the representation between multiple features and prediction, $F(x) = (F_1(x), \ldots, F_d(x))$, the representation learning to find the necessary and sufficient conditions should maximize the non-spurious and efficiency simultaneously through PNS. Given an instance x, x_i is the i-th value of x. Assuming $F_{SF} \subseteq F$ and $F_{DF} = F \setminus F_{SF}$. Based on Propositions 3 and 4, the maximization formula can be expressed as follows:

$$\max_{F_i} \sum_{i=1}^{n} \sum_{j=1}^{d} log PNS_{F_{SF}(x)=F_{SF}(x_i),T=t|F_{DF}(x)=F_{DF}(x_i)} \tag{8}$$

According to Eq.(8), to maximize the probability of PNS, it is necessary to find F_{SF}, which is the most decisive factor for the value of T. The value of F_{SF} can determine the value of T, and the value of F_{DF}, as a condition, need not affect the value of T. According to the previous analysis, the feature contained in SF is important to affect T. Under the condition of abandoned feature DF, the change in SF can determine the value of T without being affected by DF. The proof of Proposition 5 is complete.

Therefore, we should bring SF into the learning goal and generate counterfactual instances within SF.

Problem 1: *Why can we get counterfactual results by changing the value of SF?*

Consider an original instance x and a target variable of 0. We expect to reverse the target variable from 0 to 1. Therefore, we seek a counterfactual instance x^{cf} with the target value of 1 based on x.

Proposition 6. By optimizing and operating the value of SF of x, we can obtain the counterfactual instance of x with the target of 1.

Proof 3. Suppose we cannot obtain the desired target variable, i.e., in the optimization process, any value of *SF* cannot make $T = 1$,

$$P(T = 1|do(SF)) = 0 \tag{9}$$

According to causal invariance, causality does not change with context or domain. Therefore, if the causality of the target variable is found, the value of the target variable can be modified accordingly to the change in causal features. Eq.(9) shows that any operation to *SF* cannot reverse the target variable, indicating that the features in *SF* are irrelevant to the target variable. This is contrary to our Proposition 5 and Eq.(8)does not match the properties of the *MB* features contained in the *SF*; therefore, the hypothesis does not hold. The proof of Proposition 6 is complete.

Therefore, we can solve the original instance to obtain the opposite prediction by optimizing the values of features in *SF* to generate counterfactual instances. In summary, the selected feature set *SF*, as the result of the MFB, can be mined by Algorithm 1 to limit and shrink the original dataset in CEG_{MFB} so that counterfactual modifications occur only within *SF* to prevent a large number of unnecessary changes.

4.2.2. CEG_{MFB} Algorithm

In this section, we intend to generate counterfactual instances for D^* datasets with n features, which is a subset of the original dataset D and only includes causal features in *SF* mined by minFB algorithm. For each instance $x \in \mathbb{R}^n$ on D^*, its classification result $y \in \{0, 1\}$ is predicted by prediction function $p(.)$ on classifier h.

Counterfactual Explanation. Counterfactual explanation returns the corresponding counterfactual instance x^{cf} when given an original instance x. The original and counterfactual predictions are opposite, and the set δ^* indicates the difference between x^{cf} and x to generate desired reversion. For the reversion, we expect minimum δ^* for few instance changes.

Counterfactual Objectives. For x and y, we intend to obtain a counterfactual instance x^{cf} as close to x as possible where $p(x^{cf})$ is equal to a new target y^{cf} that reverse y. Therefore, the basic counterfactual objectives contain two parts, where the Part I pushes the prediction to the opposite result, and the Part II keeps the counterfactual instance close to the original instance. The general optimization objective form for generating counterfactual instances is as follows [9]:

$$\arg\min_{x^{cf}} \max_{\lambda} \lambda (p(x^{cf}) - y^{cf})^2 + d(x, x^{cf}) \tag{10}$$

where $d(.)$ is a distance function that measures how far the x^{cf} and x are from one another. λ is maximized by iteratively solving for x^{cf} and increasing λ until a sufficiently close solution is found.

Objective Function. In Part I, Eq.(10) uses the square-loss to measure the distance between $p(x^{cf})$ and y^{cf} which results in excessive punishment because a valid counterfactual only requires that $p(x^{cf})$ be greater or lesser than the threshold of $p(.)$ and does not need to be closest to the desired 1 or 0. Therefore, we use the hinge-loss instead of the square-loss in Eq.(10) to avoid large unfeasible changes of x to the counterfactual direction. Specifically, the hinge-loss ℓ is:

$$\ell = max(0, 1 - z * score(p(x^{cf}))) \tag{11}$$

where $z = -1$ when $y^{cf} = 0$ and $z = 1$ when $y^{cf} = 1, score(p(.))$ represents the classification score of $p(.)$.

Hence, the Part I of objective function is:

$$\arg\min \ell(p(x^{cf}), y^{cf}) \tag{12}$$

In Part II, we design a distance function to obtain the difference of x^{cf} and x. For continuous features, we divide each feature distance by the Median Absolute Deviation (MAD) of the feature values in the training set, because the MAD makes the indicator more reliable for measuring features that vary greatly than the more usual standard deviation. For categorical features, if any categorical feature value of the counterfactual instance is different from the original instance, assign a distance of 1; otherwise, assign 0. Specifically, the distance function is:

$$dist(x, x^{cf}) = \begin{cases} \sum_{p \in F} \frac{|x_p^{cf} - x_p|}{MAD_p} & p\ is\ continuous\ feature \\ \sum_{p \in F} I(x_p^{cf} \neq x_p) & p\ is\ categorical\ feature \end{cases} \tag{13}$$

Hence, the Part II of objective function is:

$$\arg\min dist(x, x^{cf}) \tag{14}$$

Based on the above analysis, we consider the following objective function:

$$\delta^* \leftarrow \arg\min \ell(p(x^{cf}), y^{cf}) + dist(x, x^{cf}) \tag{15}$$

The proposed two-stage CEG_{MFB} algorithm is presented in Algorithm 1. In the first stage, which is steps 1–3, we use minFB to mine the *SF* and discard other features. In the second stage, which is steps 4–9, we perform optimization within the *SF* to generate counterfactual instances.

Algorithm 2: The CEG_{MFB} Algorithm

Input: The dataset D;The feature set F;The target variable T;The add-factor θ;An instance $x = \{x_1, \ldots, x_n\}$;
Output: A counterfactual instance: x^{cf};
1./* Stage 1: find the minimal feature boundary */
2.$SF = minFB(D, F, T, \theta)$;
3. D^*: the dataset defined on SF;
4. /* Stage 2: generate counterfactual instances */
5.Train a classifier h on D^*, with prediction function p;
6. Label y using the prediction function p: $y \leftarrow p(x)$;
7.The desired class y^{cf}: the opposite of y;
8.Optimize the objective function: $\delta^* \leftarrow \arg\min \ell(p(x^{cf}), y^{cf}) + dist(x, x^{cf})$;
9.**return** $x^{cf} = x + \delta^*$;

5. Experimental Studies

For the proposed minFB and CEG_{MFB} algorithms to mine MFB and generate counterfactual instances using causal features in the MFB, respectively, a series of experiments were conducted to answer the following questions:

- **RQ1:** Is minFB effective in mining causal features? How does the add-factor θ affect the performance of minFB in mining MFB?
- **RQ2:** Are the features in MFB more outstanding than all features when using CEG_{MFB} to generate counterfactual instances?
- **RQ3:** Does CEG_{MFB} outperform other state-of-the-art counterfactual explanation generation algorithms?

5.1. General Settings

Datasets. To compare the performance of the proposed minFB, and CEG_{MFB} with their compared algorithms, respectively, we use eight real-world datasets and eight synthetic datasets. For real-world datasets from different domains, the binary class labels are used as the target variables in the experiment. Synthetic datasets are generated by a certain data generation mechanism, and we select one binary variable as the target variable. These datasets contain different numbers of continuous and categorical features. Table 2 shows the datasets used in this paper. For all datasets, we transform categorical features using one-hot-encoding, and continuous features are scaled between -1 and 1 [9]. The following metrics are calculated based on the feature value after scale. To obtain an ML model for the explanation, we use the TensorFlow library to train a non-linear neural network model with a single hidden layer.

Evaluation Metrics. The validity, proximity, sparsity, and distance are the performance metrics for counterfactual explanation. The validity measures the proportion of generated valid counterfactual instances. The proximity answers how much change is required to reach a counterfactual state. The sparsity measures the number of feature changes that distinguish the counterfactual instance from the original instance. The distance answers how far the generated counterfactual instance is from the original instance using the L_1 distance. The formulas for the evaluation metrics are shown separately below.

$$validity = \frac{n(x_v^{cf})}{n(x^{cf})} \tag{16}$$

where $n(x_v^{cf})$ represents the number of valid counterfactual instances, $n(x^{cf})$ represents the the number of all generated instances. In this paper, the value of validity is expressed as a percentage.

Table 2
Summary of the experimental datasets. Num-feature, Num-cat, and Num-cont indicate the number of all, categorical, and continuous features, respectively.

synthetic dataset				real-word dataset			
Dataset	Num-feature	Num-cat	Num-cont	Dataset	Num-feature	Num-cat	Num-cont
child	19	19	0	Prostate	5966	0	5966
alarm	36	36	0	madelon	500	0	500
andes	222	222	0	reged1	999	0	999
pathfinder	108	108	0	Divorce	54	54	0
hailfinder	55	55	0	spect	22	22	0
link	723	723	0	colon	2000	2000	0
water	31	31	0	german	20	17	3
win95pts	75	75	0	Adult	14	8	6

$$continuous - proximity(x, x^{cf}) = \frac{1}{n_{cont}} \sum_{p=1}^{n_{cont}} \frac{|x_p^{cf} - x_p|}{MAD_p} \tag{17}$$

$$categorical - proximity(x, x^{cf}) = \frac{1}{n_{cat}} \sum_{p=1}^{n_{cat}} I(x_p^{cf} \neq x_p) \tag{18}$$

where n_{cont} and n_{cat} are the number of continuous and categorical features respectively, and MAD_p is the Median Absolute Deviation from the median for the p_{th} continuous variable.

$$sparsity(x, x^{cf}) = \sum_{p=1}^{d} 1_{\{|x_p^{cf} - x_p| > t_p\}} \tag{19}$$

where t_p is a threshold that measures whether a feature has true variation.

$$distance(x, x^{cf}) = \|x^{cf} - x\|_1 \tag{20}$$

Baselines. We select the DiCE, Model-approx CF, and Proto algorithms in the category of access gradient and DiCE-Genetic, DiCE-KDTree, and DiCE-Random as representative of the black-box algorithms. Due to limitations in model type, we do not choose algorithms that access complete model internals as competitive algorithms. To verify the performance of using CEG_{MFB} to generate counterfactual instances in MFB, for comparison, we call the "Stage2: generate counterfactual instances" of CEG_{MFB} as CFBase algorithm, which uses all the features instead of those in MFB. Table 3 gives brief descriptions of all comparison methods. We use the Adam optimizer implementation in TensorFlow (learning rate = 0.05) to minimize the loss and obtain the counterfactual instances.

5.2. Experimental Results

5.2.1. Performance Analysis of minFB Algorithm (RQ1)

(1) Performance comparison of minFB and compared algorithms in mining causal features

To evaluate the performance of minFB in mining causal features, we compare the classification accuracy of mined causal features using minFB, HITON-MB, PCMB, STMB, MMMB, and BAMB, as depicted in Fig. 4. The results show that minFB has the highest classification accuracy on all datasets, indicating that it is more accurate than the compared algorithms in mining causal features of the target variable T.

Furthermore, we compare minFB with other mining causal feature methods, such as HITON-MB, PCMB, STMB, MMMB, and BAMB by using them as the mining strategies of Stage 1 in CEG_{MFB}. Because the generated counterfactual instances need to be valid, we use validity as the evaluation metric, as depicted in Fig. 5. Compared with the erratic performance of HITON-MB, PCMB, STMB, MMMB, and BAMB, the validity of counterfactual instances generated by causal features reaches 100% on 15 datasets and is higher than 90% on the Adult dataset when minFB is used as the mining strategy.

The experimental results show that minFB outperforms the compared algorithms in mining causal features. This is because minFB enhances the validity of the counterfactual algorithm by introducing the add-factor θ to compensate for causal features that may be missing. As a result, this supports the idea that we cannot only use MB as the MFB and also argues the significance of introducing the add-factor θ.

(2) Effectiveness of the add-factor θ

To verify the effectiveness of the add-factor θ, we first present the number of features in each dataset and the corresponding feature number of the MFB mined using the minFB algorithm, as shown in Table 4. Then, we demonstrate the impacts on the performance of CEG_{MFB} under different values of the add-factor θ, as shown in Tables 5 and 6. From Table 4, Figs. 8 and 9, compared with the full feature space, the MFB discards a large number of features without losing classification accuracy. In

Table 3
Summary of counterfactual explanation generation algorithms.

Access to	Algorithms	Descriptions
gradients	DiCE [14]	A framework is able to generate diverse counterfactuals for a given input for any machine learning model.
	Proto [15]	It is a model agnostic method for finding interpretable counterfactual explanations by using class prototypes.
	Model-approx CF [16]	It provided a general framework to tackle the issue of feasibility in counterfactual explanations.
only the prediction function	DiCE-Random [14]	It uses the random method to initialize the optimizer to solve the optimization problem.
	DiCE-Genetic [14]	It uses the genetic method to initialize the optimizer to solve the optimization problem.
	DiCE-KDTree [14]	It uses the KD-tree method to initialize the optimizer to solve the optimization problem.

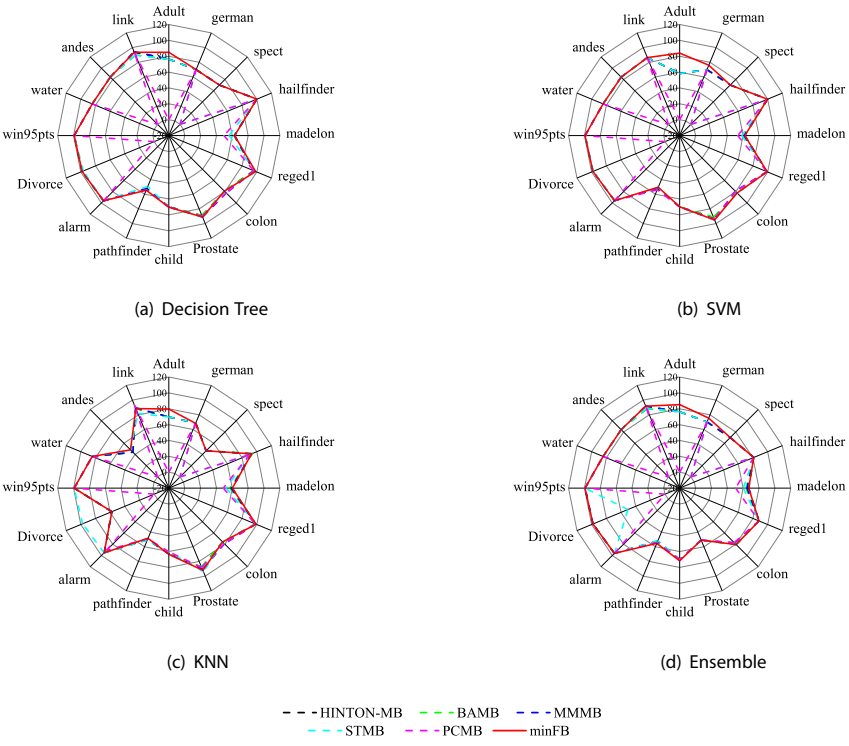


Fig. 4. Comparison of classification accuracy of causal features mined by minFB and its compared algorithms.

Tables 5 and 6, the bold lines represent the optimal auto-selected add-factor θ. The results demonstrate that 1) the optimal add-factor θ can significantly boost the validity when *MB* is insufficient to satisfy the requirements, i.e., when the add-factor θ is 0 and the validity cannot reach 100%; 2) excessively adding new features with the add-factor θ may decrease validity. Meanwhile, validity will not be increased by more iterations if it does not increase after three consecutive iterations. As presented in Table 5, the validity of the Adult dataset is 91.5%, the values of validity for the next three iterations (5–7) are less than 91.5%, and the next iterations (8–10) will also be less than 91.5%. This is because the MFB has mined sufficient causal features to accurately predict the target variable when the features introduced by the add-factor θ with high MI complete the *MB*'s missing features. Therefore, our algorithm terminates the iterative of add-factor θ on time. Tables 5 and 6 show that the iteration termination condition to iteratively generate the add-factor θ is effective with the highest validity on every dataset; 3) even if the validity remains 100% after adding new features using the add-factor θ, the evaluation metrics of proximity, sparsity, and distance increase as the number of features increase. This is because these metrics tend to be positively correlated with the add-factor θ value, which indicates that the increasing number of features will increase the complexity of generating counterfactual instances.

The above results demonstrate the importance of reducing the range of features for optimization and the necessity for counterfactual explanation to find the MFB. In addition, our add-factor θ selected by minFB possess lower proximity, sparsity, and distance values of CEG_{MFB} while ensuring the validity, which verifies the correctness of our add-factor θ selection strategy. As a result, the add-factor θ can optimize the MFB, which enables CEG_{MFB} to find counterfactual instances with low costs and high validity.

5.2.2. Comparison Analysis Using MFB and Not in CEG_{MFB} Algorithm (RQ2)

To verify the effectiveness of MFB, we compare CFBase with all features and CEG_{MFB} with MFB. Figs. 6 and 7 compare CEG_{MFB} and CFBase on the real-world/synthetic datasets, respectively.

From Figs. 6 and 7, in terms of validity, CEG_{MFB} outperformed CFBase on the 16 datasets (including real-world and synthetic datasets). Meanwhile, CEG_{MFB} has lower proximity, sparsity, and distance values. This indicates that CEG_{MFB} can find counterfactual instances at a lower cost without losing validity compared with CFBase. In addition, the validity performance of CFBase is highly erratic on different datasets, i.e., CFBase's ability to find counterfactual instances is limited by interference from irrelevant and redundant features in the dataset. In contrast, CEG_{MFB} can achieve high validity on all datasets. We conclude that removing irrelevant and redundant features and substituting all features with MFB can enhance the counterfactual explanations of CEG_{MFB}.

Furthermore, we compared the classification accuracy of all features with that of minFB, as shown in Figs. 8 and 9. The results indicate that the classification accuracy of minFB is equal to or higher than that of all features on most of the datasets,

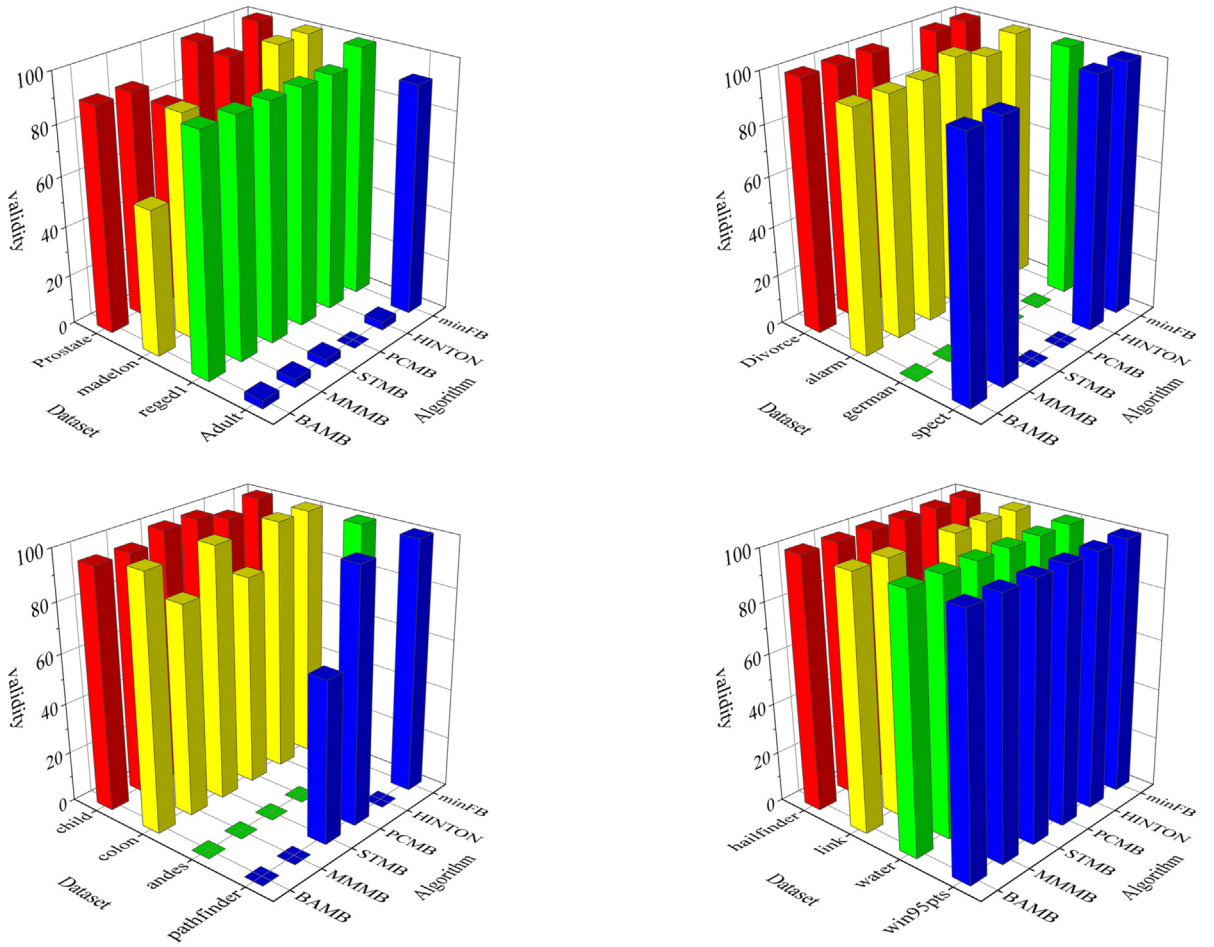


Fig. 5. Comparison of the validity(%)(↑) of counterfactual instances under the BAMB, MMMB, STMB, PCMB, HITON-MB, and minFB as mining strategies, respectively.

Table 4
Comparison of the feature number of all features and that of in MFB for every dataset. For MFB, (·) represents the feature number added by the auto-selected θ in MFB.

synthetic dataset			real word dataset		
Dataset	all features	features in MFB(·)	Dataset	all features	features in MFB(·)
child	19	6(2)	Prostate	5966	7(1)
alarm	36	5(2)	madelon	500	7(0)
andes	222	2(1)	reged1	999	12(2)
pathfinder	108	2(1)	Divorce	54	1(0)
hailfinder	55	16(0)	spect	22	1(0)
link	723	29(0)	colon	2000	2(0)
water	31	1(0)	german	20	10(8)
win95pts	75	1(0)	Adult	14	5(4)

with little difference on the remaining datasets, which means that minFB significantly ensures the classification accuracy of the target variable with the exclusion of irrelevant or redundant features and avoids spurious instances due to classification errors during optimization. This also shows that the MFB can replace all features to classify the target variable and then generate counterfactual instances.

5.2.3. Comparison Analysis CEG_{MFB} and Compared Algorithms on Generating Counterfactual Instances (RQ3)

In this section, we compare the CEG_{MFB} with six compared algorithms, as shown in Table 3 based on the evaluation metrics in Section 5.1. Tables 7–11 present the results of five evaluation metrics, respectively, including validity, continuous-

Table 5
Comparison of validity(%)(↑), sparsity(↓), distance(↓), and proximity(↓) on different add-factors θ for datasets with continuous and categorical features. The cont/cat-proximity indicates continuous/categorical-proximity, respectively. The bold lines indicate the auto-selected add-factor θ by minFB. The symbol "/" means inapplicable.

Dataset	θ	validity	sparsity	distance	proximity	
					cont-proximity	cat-proximity
Adult	0	3.9	1	5	0	0.125
	1	3.9	2	5.788	1.622	0.125
	2	70.4	2.732	4.02	1.569	0.217
	3	21.2	2.519	5.535	1.587	0.19
	4	**91.5**	**1.997**	**1.758**	**1.73**	**0**
	5	90	2.881	2.194	1.994	0
	6	82.8	2.827	2.205	2	0
	7	73.8	3.027	2.296	2.101	0
	8	75.6	3.118	2.289	2.093	0
	9	78.6	3.868	2.539	2.772	0
	10	83.3	4.836	3.498	2.909	0
german	0	0	/	/	/	/
	1	0	/	/	/	/
	2	0	/	/	/	/
	3	0	/	/	/	/
	4	0	/	/	/	/
	5	0	/	/	/	/
	6	96	3.92	4.84	3.06	0.07
	7	98	4.48	6.4	3.17	0.1
	8	**100**	**4.49**	**5.83**	**3.23**	**0.1**
	9	100	4.58	6.39	3.04	0.11
	10	93	4.96	6.56	3.16	0.13

proximity, categorical-proximity, sparsity, and distance. Once the algorithm has a validity value of 0 on a dataset, its other corresponding metrics will be shown as "/", which means inapplicable.

- **validity.** As shown in Table 7, CEG_{MFB} is more prominent than the compared algorithms. Among the compared algorithms, Proto performs better, but only above CEG_{MFB} on the Adult dataset and has 0 validity for the andes and win95pts datasets, because Proto relies on the class prototype and the counterfactual explanation is limited when the dataset does not effectively find the class prototype. DiCE achieves more than 95% validity on 7 datasets but performs poorly on the remaining datasets. Model-approx CF has over 90% validity on 8 datasets but 0 on the andes, link, and win95pts datasets. In addition, DiCE-Genetic, DiCE-KDTree, and DiCE-Random methods can achieve more than 90% validity on 6 datasets, showing that their performance is not good. In contrast, CEG_{MFB} has the best performance in terms of validity. CEG_{MFB} achieves 100% validity on 15 datasets while its validity on Adult also exceeds 90%. The reason is that CEG_{MFB} removes a large number of features that would have a spurious effect on the prediction and avoids the generation of false counterfactual instances.
- **proximity.** We measure continuous/categorical features in datasets by continuous/categorical-proximity, respectively, as shown in Table 8 with 5 datasets and Table 9 with 13 datasets. From Table 2, Adult and german contain both categorical and continuous features. In contrast, the other 14 datasets have only continuous or categorical features. For the 14 datasets, CEG_{MFB} maintains the lowest proximity. On Adult and german, CEG_{MFB}'s continuous-proximity is only higher than that of Proto when its categorical-proximity is the lowest. The other five algorithms, including DiCE-Genetic, DiCE-KDTree, DiCE-Random, and Proto can relatively achieve lower proximity. There is little difference between DiCE and Model-approx CF when the validity is not 0. The reason is that CEG_{MFB} limits the generation space of counterfactual instances and seeks to reverse the predicted changes in causal features, avoiding many unnecessary changes.
- **sparsity.** From Table 10, besides the DiCE-Random algorithm slightly outperforming CEG_{MFB} on only the Adult and german datasets, CEG_{MFB} presents significant advantages over the compared algorithms. DiCE-Random and Proto perform relatively well in terms of sparsity, whereas DiCE and Model-approx CF exhibit the worst performance. The reason is that CEG_{MFB} mines the MFB and reduces the number of features that need to be changed before finding counterfactual instances.
- **distance.** As shown in Table 11, CEG_{MFB} significantly surpasses the four algorithms, i.e., Model-approx CF, DiCE-Genetic, DiCE-KDTree, and Proto, on all datasets. Overall, CEG_{MFB} maintains the lowest distance on 13 datasets and does not differ significantly from that of the DiCE-Random algorithm on the Adult, german, and pathfinder datasets. In addition, in terms of distance, CEG_{MFB} is exponentially less than the other algorithms. The reason for this is the same as that of proximity. Then, when the validity is not 0, the distance of DiCE and Model-approx CF is higher than those of the other four algorithms, namely, DiCE-Genetic, DiCE-KDTree, DiCE-Random, and Proto.

  Above all, we observe that 1) DiCE and Model-approx CF are unstable on different datasets due to unreliable identification of causal features, resulting in the generated counterfactual instances being disturbed by irrelevant and redundant features, which reduces the validity and increases cost. By utilizing the class prototype, Proto performs well compared with

Table 6
Comparison of validity(%)(↑), sparsity(↓), proximity(↓), and distance(↓) on different add-factors θ for datasets with only continuous or categorical features. The bold lines indicate the auto-selected add-factor θ by minFB. For proximity, the datasets of Prostate, madelon, and reged1 use continuous-proximity due to include only continuous-features, while other datasets use categorical-proximity. The symbol "/" means inapplicable.

Dataset	θ	validity(%)	proximity	sparsity	distance	Dataset	θ	validity(%)	proximity	sparsity	distance
alarm	0	94.8	0.05	1.67	2	Divorce	**0**	**100**	**0.02**	**1**	**2.53**
	1	99.4	0.07	2.38	2.69		1	88.2	0.03	1.73	2.47
	2	**100**	**0.08**	**2.87**	**4.13**		2	100	0.02	1.29	2.88
	3	100	0.12	4.24	6.84		3	100	0.03	1.41	4.35
	4	100	0.12	4.4	7.1		4	100	0.03	1.41	4.82
	5	100	0.12	4.36	7.08		5	100	0.03	1.47	4.93
spect	**0**	**100**	**0.05**	**1**	**1**	child	0	98	0.06	1.37	1.78
	1	100	0.08	1.78	1.78		**1**	**100**	**0.05**	**1.08**	**1.64**
	2	96.3	0.09	1.92	1.92		2	100	0.05	1.16	1.68
	3	100	0.05	1.19	1.19		3	100	0.06	1.32	1.6
	4	100	0.07	1.63	1.63		4	100	0.07	1.58	2.3
	5	100	0.12	2.67	2.67		5	100	0.08	1.78	2.74
colon	**0**	**100**	**0**	**1**	**2.33**	andes	0	0	/	/	/
	1	100	0	1	2.33		**1**	**100**	**0.01**	**1.32**	**1.32**
	2	100	0	1	2.33		2	100	0.01	1.36	1.36
	3	100	0	1.5	4		3	44	0.01	1.45	1.45
	4	100	0	3.33	8		4	100	0.01	2.22	2.22
	5	100	0	3	7.33		5	100	0.01	1.78	1.78
pathfinder	0	2	0.01	1	2	hailfinder	**0**	**100**	**0.06**	**3.54**	**13.06**
	1	**100**	**0.01**	**1.46**	**11.34**		1	100	0.07	3.68	14.4
	2	100	0.02	2.36	38.58		2	100	0.07	3.6	13.7
	3	100	0.02	2.64	39.76		3	100	0.07	3.72	14.64
	4	100	0.02	2.42	38.78		4	100	0.07	3.84	15.48
	5	100	0.03	3.44	41.42		5	100	0.07	3.84	15.28
link	**0**	**100**	**0**	**1.3**	**2.32**	water	**0**	**100**	**0.03**	**1**	**1**
	1	100	0	1.3	2.36		1	100	0.03	1	1
	2	100	0	1.34	2.42		2	100	0.03	1	1
	3	100	0	1.38	2.46		3	100	0.03	1	1
	4	100	0	1.34	2.38		4	100	0.06	1.76	2.42
	5	98	0	1.43	2.45		5	100	0.06	1.84	2.5
win95pts	**0**	**100**	**0.01**	**1**	**1**	Prostate	0	90	0	5.89	3.7
	1	100	0.01	1	1		**1**	**100**	**0**	**6.8**	**4.14**
	2	100	0.01	1	1		2	100	0	7.7	4.26
	3	100	0.01	1	1		3	100	0	8.7	4.79
	4	100	0.01	1	1		4	90	0	9.78	5.28
	5	100	0.01	1	1		5	70	0	10.71	6.05
madelon	**0**	**100**	**0.01**	**5.96**	**1.44**	reged1	0	94	0.07	9.96	6.45
	1	100	0.02	6.4	1.87		1	94	0.08	10.96	6.86
	2	100	0.02	7.03	1.82		**2**	**100**	**0.08**	**11.94**	**7.07**
	3	100	0.02	7.86	1.96		3	100	0.08	12.94	7.44
	4	100	0.02	8.48	2.08		4	100	0.08	13.92	7.8
	5	100	0.02	9.15	2.26		5	100	0.08	14.92	8.25

the other algorithms, excluding CEG_{MFB}. However, due to the lack of causal constraints, the number of features is also large, which denotes a huge computation cost. DiCE-Genetic, DiCE-KDTree, and DiCE-Random achieve lower proximity, sparsity, and distance at the cost of validity. 2) Overall, our CEG_{MFB} is the most efficient and achieves the best performance among all algorithms. CEG_{MFB} improves validity; conversely, it reduces proximity, sparsity, and distance. The experimental results show that compared with other algorithms, the counterfactual instances generated by CEG_{MFB} changes less without sacrificing validity, which is consistent with the two basic properties of counterfactual explanation. This is because CEG_{MFB} mines the MFB and reduces the range of features for optimization, making the target variable find counterfactual instances only on important causal features and reversing the results with effective changes, which reduces the cost while excluding the erroneous influence of irrelevant/redundant features to avoid the generation of a large number of false counterfactual instances and improve the validity.

5.2.4. Feature Changes in Generated Counterfactual Instances

To observe the changes of features in MFB, we use 16 datasets to generate counterfactual instances by CEG_{MFB} and count the change ratio (CR) of each feature in the MFB, respectively. $CR(F_i)$ is the change ratio of feature F_i on counterfactual instances, $CR(F_i)$=$nCI(F_i)$/nTI, where nCI and nTI denote the number of instances changed on F_i and the number of total instances, respectively. Note that we use the change ratio of each feature instead of the number of changes because the number of instances in each dataset is different, which is unconducive to obtaining convincing statistics.

The results in Fig. 10 show that each feature in the MFB must not be modified for every original instance. Although whether a feature is modified is related to the initial value of a specific instance, the feature CR can further explain the

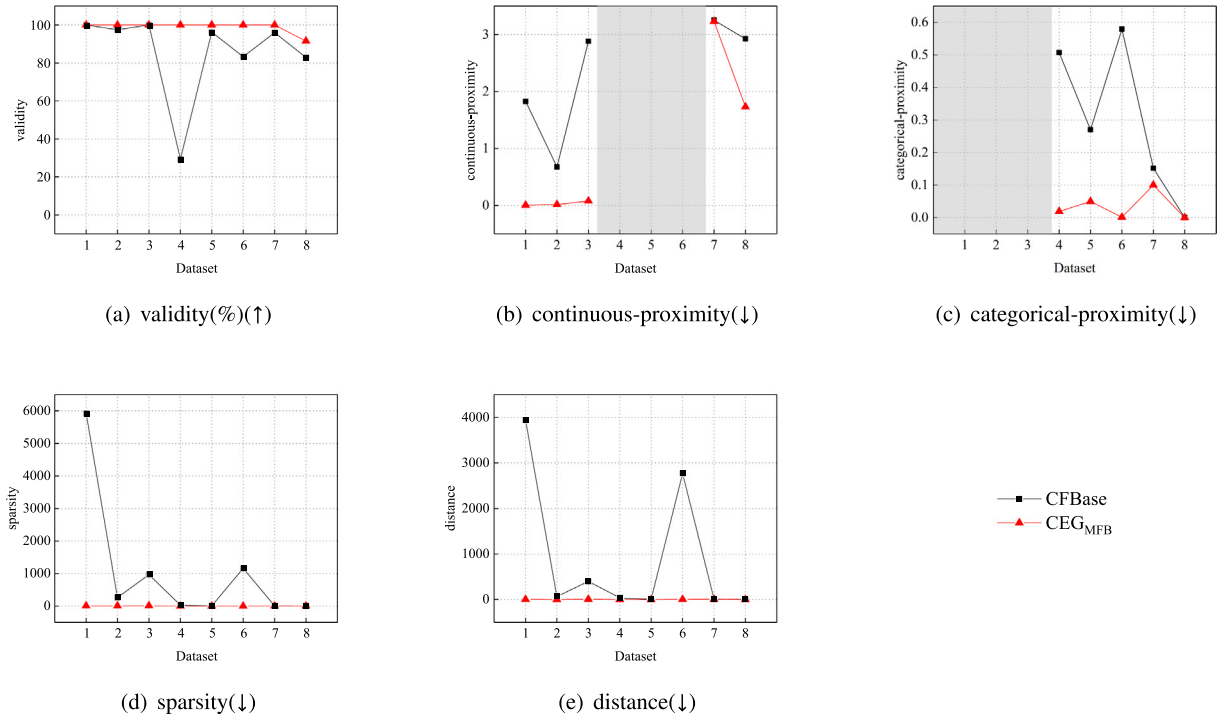


Fig. 6. Comparison of CEG_{MFB} with CFBase for various evaluation metrics on real-world datasets. The labels of the x-axis from 1 to 8 denote the datasets: 1: Prostate; 2:madelon; 3:reged1; 4:Divorce; 5:spect; 6:colon; 7:german; 8:Adult. For the datasets of 4, 5, and 6 only include categorical features, continuous-proximity is inapplicable. Similarly, the three datasets of 1, 2, and 3 only include continuous features, categorical-proximity is inapplicable, as shown in gray shading in (b) and (c).

importance of the feature in the MFB and the relationship between the feature and the target variable to some extent. For example, in the Adult dataset, the feature #11 representing "capital-gain", grows each time when the target variable changes from negative to positive which illustrates the importance of capital-gain to income. Meanwhile, it also explains the relationship between capital-gain and income, i.e., when capital-gain increase, income will also increase, which is consistent with people's cognition in real life, indicating that the counterfactual explanation is effective.

5.2.5. Summary of Experiments

Based on the above experimental results, we summarize CEG_{MFB}'s advantages and disadvantages as follows.

- **Advantages.** 1) Compared with different existing counterfactual generation algorithms, CEG_{MFB} first identifies the causal features of the target variable by mining the MFB. This makes it possible to generate counterfactual instances more accurately. Conversely, existing algorithms generate counterfactual instances with all features, resulting in low performance; 2) Compared with existing counterfactual generation algorithms, CEG_{MFB} is less affected by high feature dimensions. For datasets with vast features, CEG_{MFB} can still generate valid counterfactual instances within the limited features and make smaller changes to the original instances.
- **Disadvantages.** 1) Mining the MFB costs extra computation. However, this computation cost is acceptable because each dataset only needs to mine the MFB once. As the number of instances increases, the overall cost of generating counterfactual instances will decrease because only the MFB is executed; 2) Reversal of classification is limited to binary classification tasks. In future work, we plan to extend counterfactual explanation to multi-classification problems capable of generating counterfactual instances at specific desired classes.

5.3. A Case Study in Real Scenario

The real case comes from the National Center for Biotechnology Information (NCBI) to conduct experiments for explaining glioma grading. The dataset, named GLI, derives from 74 patients surgically treated in multiple surgical departments at the University of California, and predicts the grading of the most common types of primary malignancies found in adults. We used all patients' gliomas as examples and each department's test item as features. Before this, all prediction models and analysis methods only tried to achieve better classification and did not explain the prediction. We interpret the grades III and IV gliomas that are easy to be confused and result in a total number of 85 instances and 22,283 features for the GLI.

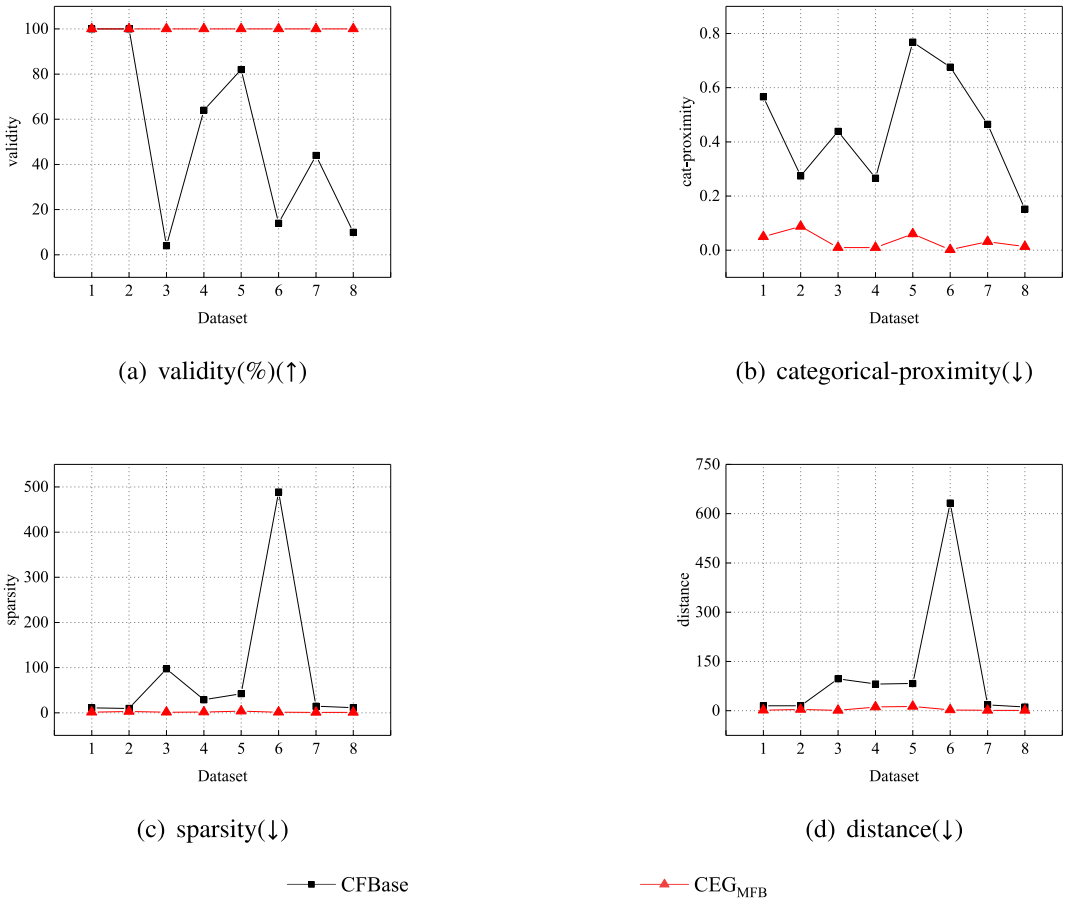


Fig. 7. Comparison of CEG_{MFB} with CFBase for various evaluation metrics on synthetic datasets. The labels of the x-axis from 1 to 8 denote the datasets: 1: child; 2:water; 3:alarm; 4:hailfinder; 5:win95pts; 6:pathfinder; 7:andes; 8:link.

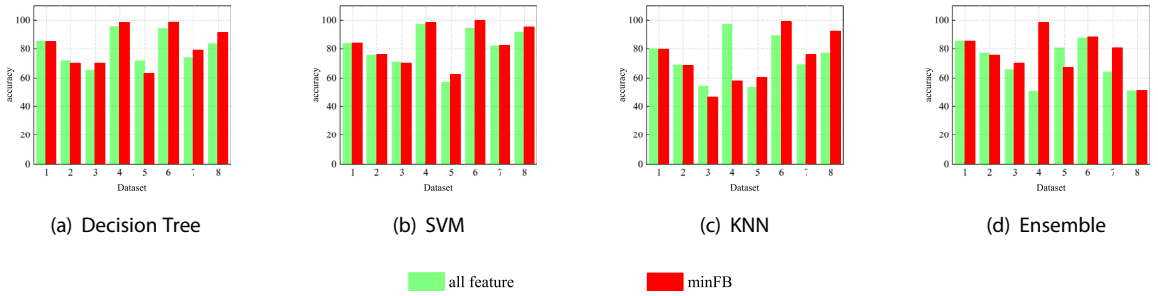


Fig. 8. Comparison of classification accuracy of all features with minFB on real-world datasets: 1:Adult; 2:german; 3:spect; 4:Divorce; 5:madelon; 6: reged1; 7:colon; 8:Prostate.

Fig. 11 shows the illustration of glioma. In Fig. 11, "-" indicates grade IV, "+" indicates grade III. Transcriptional profiling is important to elucidate glioma biology, prognosticate survival, and define tumor sub-classes. By explaining the prediction, we can help glioma classification, which means that if transcriptional profiling can't accurately judge the grade, we can pay attention to the mined MFB and its values, and then determine the grade according to the decision boundary gained on the counterfactual instance.

Performance analysis of algorithm on GLI. In the experiments, we apply the CEG_{MFB} and six counterfactual explanation generation algorithms to the scenario, and the results are presented in Table 12. Among these algorithms, 1) CEG_{MFB} generates valid counterfactual instances with the lowest proximity, sparsity, and distance compared with its compared algorithms; 2) the validity of DiCE-Genetic and DiCE-KDTree is 0, and the performance is the worst. It is closely followed by

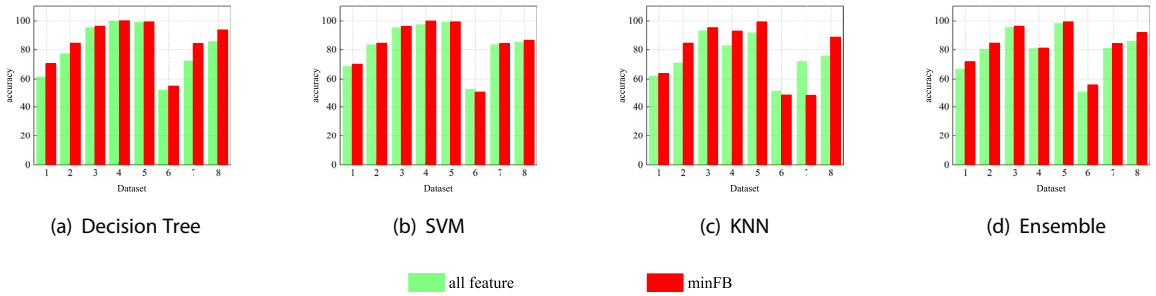


Fig. 9. Comparison of classification accuracy of all features with minFB on synthetic datasets: 1:child; 2:water; 3:alarm; 4:hailfinder; 5:win95pts; 6:pathfinder; 7:andes; 8:link.

Table 7
Comparison of the validity(%)(↑) of seven counterfactual explanation generation algorithms. The win/tie/loss counts for CEG_{MFB} are summarized in the last row, abbreviated as w/t/l. The symbol • indicates CEG_{MFB} outperforms its compared algorithms, and ○ indicates a tie.

	Algorithm						
Dataset	DiCE	Model-approx CF	DiCE-Genetic	DiCE-KDTree	DiCE-Random	Proto	CEG_{MFB}
Prostate	**100**○	**100**○	20•	20•	30•	**100**○	**100**
madelon	99.5•	85•	52.5•	53•	31•	**100**○	**100**
reged1	**100**○	94•	4•	4•	38•	**100**○	**100**
Adult	77.4•	78.3•	78.4•	78.3•	74.5•	**99.5**	91
german	99•	87•	3•	0•	65•	59•	**100**
alarm	**100**○	99•	96.8•	0•	37•	99.8•	**100**
Divorce	23.5•	**100**○	88.24•	**100**○	47.1•	**100**○	**100**
spect	96.3•	96.3•	62.9•	63•	96.3•	**100**○	**100**
child	88•	**100**○	32•	28•	98•	**100**○	**100**
colon	66.7•	50•	50•	**100**○	0•	**100**○	**100**
andes	**100**○	0•	96•	**100**○	6•	0•	**100**
pathfinder	41.2•	94.1•	58•	56•	62•	**100**○	**100**
hailfinder	82•	32•	18•	18•	84•	**100**○	**100**
link	16.7•	0•	8•	8•	14•	88•	**100**
water	36•	**100**○	40•	44•	86•	**100**○	**100**
win95pts	10•	0•	90•	90•	64•	0•	**100**
w/t/l	12/4/0	12/4/0	16/0/0	13/3/0	16/0/0	5/10/1	—

Table 8
Comparison of the continuous-proximity(↓) of seven counterfactual explanation generation algorithms. The symbol "/" means inapplicable. The win/tie/loss counts for CEG_{MFB} are summarized in the last row, abbreviated as w/t/l. The symbol • indicates CEG_{MFB} outperforms its compared algorithms, and ○ indicates a tie.

	Algorithm						
Dataset	DiCE	Model-approx CF	DiCE-Genetic	DiCE-KDTree	DiCE-Random	Proto	CEG_{MFB}
Prostate	1.828•	1.932•	2.062•	1.1•	0.•	0.243•	**0.002**
madelon	0.708•	1.261•	1.591•	1.515•	0.04•	0.386•	**0.013**
reged1	2.887•	3.263•	1.66•	1.534•	0.889•	0.793•	**0.077**
Adult	3.551•	3.233•	0.978	1.118	0.549	**0.306**	1.73
german	3.373•	3.462•	3.978•	/•	2.23	**0.838**	3.209
w/t/l	5/0/0	5/0/0	4/0/1	4/0/1	3/0/2	3/0/2	—

DiCE-Random with a validity of only 22.2% and has considerable proximity, sparsity, and distance. The reason is that DiCE-Genetic, DiCE-KDTree, and DiCE-Random methods try to change the original instances directly to find counterfactual instances, and such changes are difficult with a large number of features; 3) for DiCE, the run time is more than 10 days, which is obviously undesirable; 4) although Model-approx CF has a validity of 100%, other metrics are still deficient; 5) Proto is not comparable with CEG_{MFB} in the metrics of sparsity and distance although its validity is 100%.

Analysis of generated counterfactual instances. To observe the results of generated counterfactual instances by CEG_{MFB} on GLI, we present the changes of feature values in the MFB when generating counterfactual instances, as shown in Table 13. Table 13 lists seven random original and corresponding counterfactual instances on GLI. Note that only the features in the MFB are modified to generate counterfactual instances. The five original instances (No.1-No.5) are classified as "-", indicating level IV, and the corresponding counterfactual instances are reversed to "+", indicating level III. The other two original

Table 9
Comparison of the categorical-proximity(↓) of seven counterfactual explanation generation algorithms. The symbol "/" means inapplicable. The win/tie/loss counts for CEG_{MFB} are summarized in the last row, abbreviated as w/t/l. The symbol • indicates CEG_{MFB} outperforms its compared algorithms, and ○ indicates a tie.

	Algorithm						
Dataset	DiCE	Model-approx CF	DiCE-Genetic	DiCE-KDTree	DiCE-Random	Proto	CEG_{MFB}
Adult	**0**○	0.412•	0.343•	0.441•	0.065•	0.321•	**0**
german	0.192•	0.498•	0.274•	/•	0.138•	0.236•	**0.077**
alarm	0.283•	0.348•	0.344•	/•	0.103•	0.232•	**0.088**
Divorce	0.515•	0.514•	0.626•	0.836•	0.263•	0.583•	**0.019**
spect	0.281•	0.284•	0.106•	0.125•	0.088•	0.163•	**0.05**
child	0.576•	0.628•	0.201•	0.313•	0.106•	0.161•	**0.05**
colon	0.59•	0.697•	0.52•	0.51•	/•	0.246•	**0.001**
andes	0.447•	/•	0.306•	0.357•	0.267•	/•	**0.006**
pathfinder	0.273•	0.289•	0.163•	0.26•	0.042•	0.051•	**0.016**
hailfinder	0.769•	0.753•	0.471•	0.514•	0.18•	0.219•	**0.056**
link	0.68•	/•	0.285•	0.323•	0.166•	0.293•	**0.002**
water	0.461•	0.457•	0.157•	0.23•	0.064•	0.119•	**0.032**
win95pts	0.152•	/•	0.088•	0.186•	0.025•	/•	**0.013**
w/t/l	12/1/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	–

Table 10
Comparison of the sparsity(↓) of seven counterfactual explanation generation algorithms. The symbol "/" means inapplicable. The win/tie/loss counts for CEG_{MFB} are summarized in the last row, abbreviated as w/t/l. The symbol • indicates CEG_{MFB} outperforms its compared algorithms, and ○ indicates a tie.

	Algorithm						
Dataset	DiCE	Model-approx CF	DiCE-Genetic	DiCE-KDTree	DiCE-Random	Proto	CEG_{MFB}
Prostate	5916.4•	5919.9•	3645.5•	3946•	750.333•	2468.7•	**6.8**
madelon	274.553•	487.959•	474.743•	401.991•	6.113•	273.705•	**5.865**
reged1	969.86•	989.191•	951.5•	981.5•	196.158•	678.66•	**11.94**
Adult	4.931•	9.235•	4.612•	5.491•	**1.639**	4.816•	1.997
german	5.96•	11.448•	7.333•	/•	**3.446**	5.949•	4.11
alarm	10.164•	12.511•	12.401•	/•	3.632•	8.339•	**2.87**
Divorce	27.75•	27.765•	33.8•	45.176•	14.125•	31.471•	**1**
spect	6.192•	6.231•	2.294•	2.706•	1.962•	3.593•	**1**
child	10.955•	11.94•	3.812•	5.929•	1.939•	3.06•	**1.16**
colon	1178.25•	1394.333•	1045•	1021•	/•	491.333•	**1**
andes	99.26•	/•	67.75•	79.32•	58.333•	/•	**1.32**
pathfinder	29.429•	31.188•	17.448•	28.179•	4.419•	5.48•	**1.46**
hailfinder	42.293•	41.438•	26•	28.333•	9.833•	12.04•	**3.54**
link	492•	/•	206.25•	232.75•	130.286•	211.886•	**1.3**
water	14.278•	14.18•	4.85•	7.136•	1.977•	3.82•	**1**
win95pts	11.4•	/•	6.622•	13.956•	1.875•	/•	**1**
w/t/l	16/0/0	16/0/0	16/0/0	16/0/0	14/0/2	16/0/0	–

Table 11
Comparison of the distance(↓) of seven counterfactual explanation generation algorithms. The symbol "/" means inapplicable. The win/tie/loss counts for CEG_{MFB} are summarized in the last row, abbreviated as w/t/l. The symbol • indicates CEG_{MFB} outperforms its compared algorithms, and ○ indicates a tie.

	Algorithm						
Dataset	DiCE	Model-approx CF	DiCE-Genetic	DiCE-KDTree	DiCE-Random	Proto	CEG_{MFB}
Prostate	3941.126•	4161.254•	3149.918•	2009.599•	820.427•	438.538•	**4.144**
madelon	70.036•	123.843•	155.604•	148.201•	4.089•	37.028•	**1.44**
reged1	401.299•	466.097•	298.677•	272.43•	134.179•	146.81•	**7.07**
Adult	3.552•	7.05•	3.513•	4.365•	**1.566**	7.978•	1.758
german	5.003	10.244•	6.585•	/•	**3.458**	8.138•	5.83
alarm	14.802•	18.677•	16.442•	/•	5•	10.85•	**4.13**
Divorce	36.75•	51.588•	67.133•	105.176•	32.125•	61.529•	**2.53**
spect	6.192•	6.231•	2.294•	2.706•	1.962•	3.593•	**1**
child	16.023•	17.16•	5.062•	7.357•	2.694•	3.96•	**1.68**
colon	2754•	3468.667•	2291.333•	2312•	/•	496•	**2.333**
andes	99.26•	/•	67.75•	79.32•	58.333•	/•	**1.32**
pathfinder	86.571•	86.5•	37.103•	57.786•	**8.129**	16.9•	11.34
hailfinder	82.537•	77.562•	46.444•	52.222•	19.31•	27.54•	**13.06**
link	638•	/•	257.5•	296.75•	120.286•	271.568•	**2.32**
water	14.278•	14.18•	4.85•	7.136•	1.977•	4.52•	**1**
win95pts	11.4•	/•	6.622•	13.956•	1.875•	/•	**1**
w/t/l	15/0/1	16/0/0	16/0/0	16/0/0	13/0/3	16/0/0	–

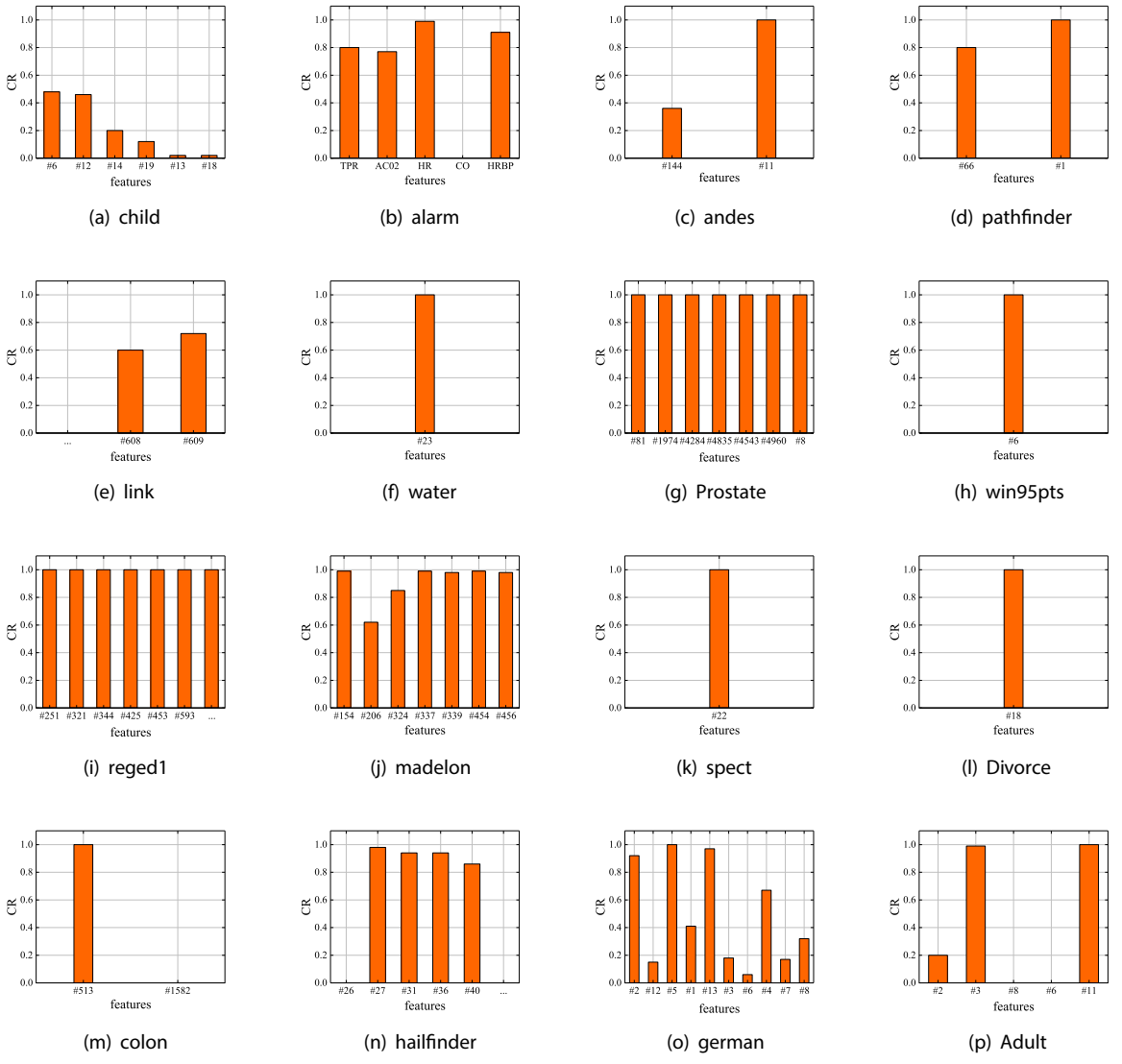


Fig. 10. The change ratio of each MFB feature in the counterfactual instances generated on the dataset. The "#" represents the ID of the features, and "..." indicates that the remaining features in MFB which are not represented on the x-axis.

instances (No.6 and No.7) are classified as "+", and the corresponding counterfactual instances are reversed to "-". Table 13 shows the decision boundary of classification, which will be summarized in the last row. Once the values of all features in the MFB remain on the same side of the decision boundary, which means that none of them is lower than (none is higher than) the value on the boundary, the instances will be classified as "+" ("-"), as shown in Fig. 12(a). However, the generated counterfactual instances will be closer to the decision boundary to obtain the minimum change in reversion classification. Therefore, to further illustrate how features change when the classification is reversed, we select instances No.2 and No.4-No.7 from Table 13, along with their corresponding counterfactual instances, as shown in Figs. 12(a) and (b). For the blue negative (red positive) original instances in Fig. 12(a), the original features above (below) the decision boundary remain constant, whereas the original features below (above) the decision boundary will change to the value on the decision boundary when the counterfactual instances are reversed, which shown as blue positive (red negative) counterfactual instances in Fig. 12(b).

To summarize, we found that CEG_{MFB} can 1) provide clear suggestions on how to change the value of a feature to achieve the desired result; 2) explain the prediction results, i.e., if the values of features on instances cross the decision boundary, it will be divided into the opposite class by the classifier; 3) help more quickly and correctly make predictions and decisions because causal features are mined in advance, especially for high-dimensional data.

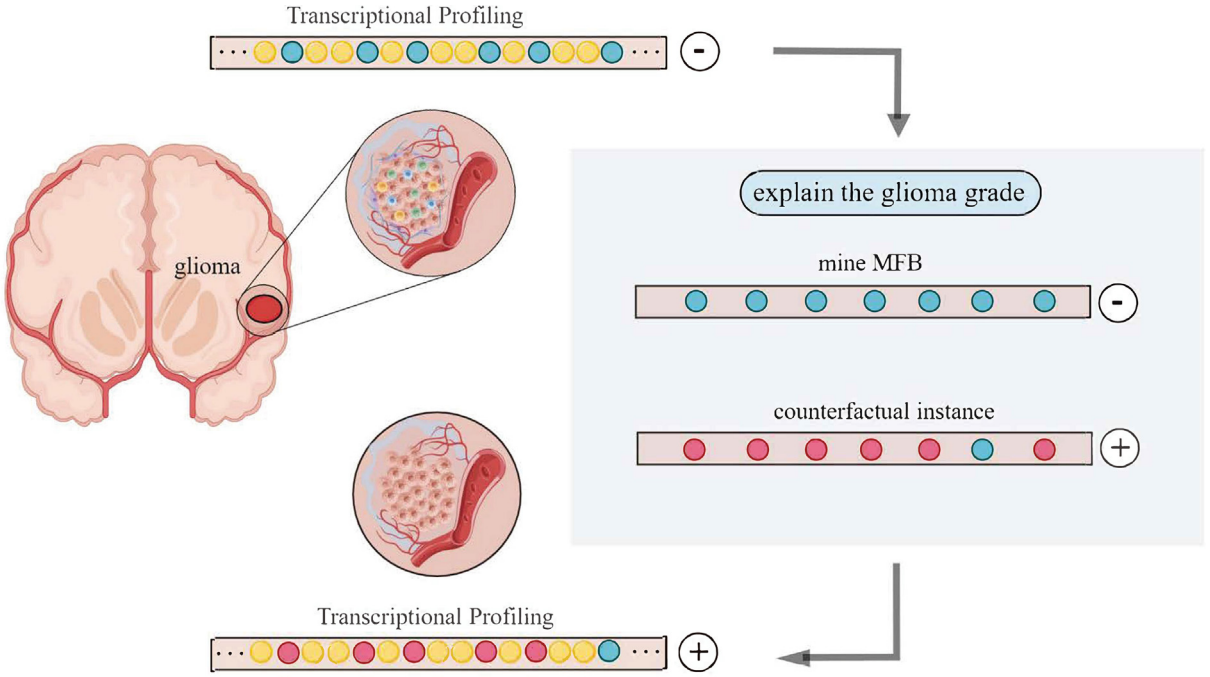


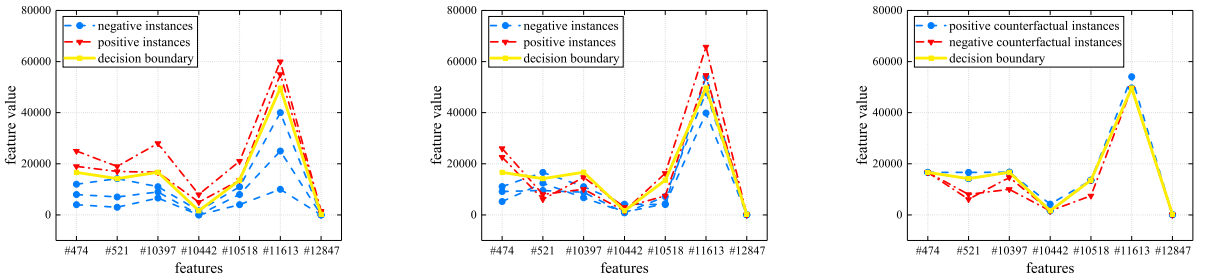
Fig. 11. The illustration of the GLI scenario. In the figure, "−" and "+" indicate grades IV and III respectively, and the blue nodes indicate the causal features in the mined MFB by minFB; The red nodes are updated causal features in MFB by the counterfactual generation algorithm CEG_{MFB} to revise the prediction of target variable from "-" to "+".

Table 12
Comparison of evaluation metrics for different algorithms and some descriptions in CEG_{MFB} for GLI scenario. The symbol '/' means inapplicable, "-" indicates undesirable.

Approach	validity(%)	continuous-proximity	sparsity	distance
DiCE	-	-	-	-
Model-approx CF	**100**	4.09	22088.222	11955.328
DiCE-Genetic	0	/	/	/
DiCE-KDTree	0	/	/	/
DiCE-Random	22.2	6246.678	7191.5	4673.13
Proto	**100**	0.869	13400.333	3686.98
CEG_{MFB}	**100**	**0.017**	**6.778**	**4.173**
Description	Number of all features: 22283; Number of features in MFB: 7; The minimal feature boundary (MFB): {474, 521, 10397, 10442, 10518, 11613, 12847};			

Table 13
Seven original instances and the corresponding generated counterfactual instances on GLI. The instances are randomly selected. The "#" represents the ID of the features in MFB, and No.* represents the order of the selected instances in the dataset. The • represents the decision boundary of features for the prediction reversion.

		features in MFB							
	instances	#474	#521	#10397	#10442	#10518	#11613	#12847	class
No.1	original	14149	9060	6507	782	3245	64974	57	-
	counterfactual	16614•	14192•	16690•	1590•	13606•	64974	72•	+
No.2	original	11069	15636	11055	1946	4036	39848	51	-
	counterfactual	16614•	15636	16690•	1946	13606•	49652•	72•	+
No.3	original	9463	5702	6602	1637	4607	40075	46	-
	counterfactual	16614•	14192•	16690•	1637	13606•	49652•	72•	+
No.4	original	5208	12394	6704	1080	4351	54082	49	-
	counterfactual	16614•	14192•	16690•	1590•	13606•	54082	72•	+
No.5	original	9147	9580	8828	798	7162	47957	41	-
	counterfactual	16614•	14192•	16690•	1590•	13606•	49652•	72•	+
No.6	original	26005	8009	10003	1579	16271	65763	104	+
	counterfactual	16614•	8009	10003	1579	13606•	49652•	72•	-
No.7	original	22575	6115	14703	2982	7457	54656	55	+
	counterfactual	16614•	6115	14703	1590•	7457	49652•	55	-
...	...	...	...	...	...	...	...	...	
	decision boundary	16614	14192	16690	1590	13606	49652	72	



(a) instances and decision boundary on GLI. (b) original instances. (c) generated counterfactual instances.

Fig. 12. Analysis of generated counterfactual instances on GLI.

6. Conclusion

In this study, we propose a new counterfactual explanation generation algorithm, namely, CEG_{MFB} with a minimal feature boundary (MFB) mined by our minFB algorithm. First, the minFB algorithm is designed to mine the MFB using the causality between features and the target variable. Second, counterfactual instances are generated by CEG_{MFB} using causal features in the MFB. For the minFB algorithm, we demonstrate the iterative process of add-factor θ and evaluate the MFB prediction performance. The experimental results indicate that the add-factor θ can further mine missing causal features and improve the prediction accuracy over compared algorithms. Moreover, we discuss the performance of the CEG_{MFB} algorithm in terms of the following evaluation metrics: validity, proximity, sparsity, and distance. The experimental results are as follows: 1) The results indicate that CEG_{MFB} significantly outperforms existing counterfactual explanation generation algorithms, including DiCE, Model-approx CF, and Proto; 2) The CEG_{MFB} algorithm can generate counterfactual instances only by modifying the features in the MFB to achieve the reversing prediction result due to the introducing of MFB; 3) The CEG_{MFB} algorithm can realize the minimum reversing cost by minimizing the distance between the original and the counterfactual instances; 4) On different real-world/synthetic datasets, CEG_{MFB} maintains high validity and robustness.

Our method explains the single prediction by generating counterfactual instances after mining causality. However, some limitations exist such as the extra costs in the minFB stage and the binary classification tasks constraint. In the future, we are committed to improving performance by solving these limitations and more research need to be followed up in the counterfactual explanation fields, such as counterfactual inference under the constraints of non-IID, multimodal data, and streaming data.

CRediT authorship contribution statement

Dianlong You: Conceptualization, Methodology, Formal analysis, Data curation, Writing - original draft, Writing - review & editing, Validation, Funding acquisition. **Shina Niu:** Conceptualization, Methodology, Data curation, Writing - original draft, Writing - review & editing. **Siqi Dong:** Investigation, Resources. **Huigui Yan:** Investigation, Resources. **Zhen Chen:** Funding acquisition, Resources. **Di Wu:** Resources, Funding acquisition. **Limin Shen:** Funding acquisition, Supervision. **Xindong Wu:** Methodology, Supervision.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been partially supported by the grants from National Natural Science Foundation of China No.62276226, No.62176070, No.62102348, and No.62120106008; Hebei Natural Science Foundation No.F2021203038, and No. F2022203012.

References

[1] R. Moraffah, M. Karami, R. Guo, A. Raglin, H. Liu, Causal interpretability for machine learning-problems, methods and evaluation, ACM SIGKDD Explorations Newsletter 22 (1) (2020) 18–33.
[2] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intelligent Systems 34 (6) (2019) 14–23.
[3] T. Wang, Q. Lin, Hybrid predictive models: when an interpretable model collaborates with a black-box model, Journal of Machine Learning Research 22 (137) (2021) 1–38.
[4] A.J. London, Artificial intelligence and black-box medical decisions: accuracy versus explainability, Hastings Center Report 49 (1) (2019) 15–21.
[5] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA), IEEE, 2018, pp. 80–89.
[6] N. Kilbertus, P.J. Ball, M.J. Kusner, A. Weller, R. Silva, The sensitivity of counterfactual fairness to unmeasured confounding, in: Uncertainty in artificial intelligence, PMLR, 2020, pp. 616–626.
[7] D. Slack, A. Hilgard, H. Lakkaraju, S. Singh, Counterfactual explanations can be manipulated, Advances in Neural Information Processing Systems 34 (2021) 62–75.
[8] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: A review, arXiv preprint arXiv:2010.10596.
[9] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Social Science Research Network electronic journal (SSRN) 31 (2017) 842–887.
[10] H.-G. Jung, S.-H. Kang, H.-D. Kim, D.-O. Won, S.-W. Lee, Counterfactual explanation based on gradual construction for deep networks, Pattern Recognition 132 (2022) 108958.
[11] J. Kim, M. Kim, Y.M. Ro, Interpretation of lesional detection via counterfactual generation, in: 2021 IEEE International Conference on Image Processing (ICIP), IEEE, 2021, pp. 96–100.
[12] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st international conference on neural information processing systems, 2017, pp. 4768–4777.
[13] A. Datta, S. Sen, Y. Zick, Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems, in: 2016 IEEE symposium on security and privacy (SP), IEEE, 2016, pp. 598–617.
[14] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 607–617.
[15] A.V. Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 650–665.
[16] D. Mahajan, C. Tan, A. Sharma, Preserving causal constraints in counterfactual explanations for machine learning classifiers, arXiv preprint arXiv:1912.03277.
[17] R.R. Fernández, I.M. De Diego, V. Aceña, A. Fernández-Isabel, J.M. Moguerza, Random forest explainability using counterfactual sets, Information Fusion 63 (2020) 196–207.
[18] J. Kaddour, A. Lynch, Q. Liu, M.J. Kusner, R. Silva, Causal machine learning: A survey and open problems, arXiv preprint arXiv:2206.15475.
[19] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (5) (2018) 1–42.
[20] H. Deng, Interpreting tree ensembles with intrees, International Journal of Data Science and Analytics 7 (4) (2019) 277–287.
[21] V. Guyomard, F. Fessant, T. Bouadi, T. Guyet, Post-hoc counterfactual generation with supervised autoencoder, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 105–114.
[22] C. Molnar, G. Casalicchio, B. Bischl, Interpretable machine learning–a brief history, state-of-the-art and challenges, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 417–431.
[23] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.
[24] M.T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[25] C. Fernández-Loría, F. Provost, X. Han, Explaining data-driven decisions made by ai systems: the counterfactual approach, arXiv preprint arXiv:2001.07417.
[26] J. Pearl, D. Mackenzie, The book of why: the new science of cause and effect, Basic Books New York, 2019.
[27] R.M. Byrne, Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning, in: IJCAI International Joint Conference on Artificial Intelligence, Vol. 2019-August, 2019, pp. 6276–6282.
[28] K. Mohammadi, A.-H. Karimi, G. Barthe, I. Valera, Scaling guarantees for nearest counterfactual explanations, in: Proceedings of the 2021 AAAI/ACM Conference on AI, and Society, 2021, pp. 177–187.
[29] K. Kanamori, T. Takagi, K. Kobayashi, H. Arimura, Dace, Distribution-aware counterfactual explanation by mixed-integer linear optimization, in: IJCAI International Joint Conference on Artificial Intelligence, 2021, pp. 2855–2862.
[30] A.-H. Karimi, G. Barthe, B. Balle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 895–905.
[31] A.-H. Karimi, B. Schölkopf, I. Valera, Algorithmic recourse: from counterfactual explanations to interventions, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 353–362.
[32] C. Russell, Efficient search for diverse coherent explanations, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 20–28.
[33] J. Bien, R. Tibshirani, Prototype selection for interpretable classification, The Annals of Applied Statistics 5 (4) (2011) 2403–2424.
[34] R.M. Grath, L. Costabello, C.L. Van, P. Sweeney, F. Kamiab, Z. Shen, F. Lecue, Interpretable credit application predictions with counterfactual explanations, arXiv preprint arXiv:1811.05245.
[35] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Comparison-based inverse classification for interpretability in machine learning, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2018, pp. 100–111.
[36] A. Dhurandhar, T. Pedapati, A. Balakrishnan, P.-Y. Chen, K. Shanmugam, R. Puri, Model agnostic contrastive explanations for structured data, arXiv preprint arXiv:1906.00117.
[37] S. Dandl, C. Molnar, M. Binder, B. Bischl, Multi-objective counterfactual explanations, in: International Conference on Parallel Problem Solving from Nature, Springer, 2020, pp. 448–469.
[38] K. Yu, X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, X. Wu, Causality-based feature selection: Methods and evaluations, ACM Computing Surveys (CSUR) 53 (5) (2020) 1–36.
[39] K. Yu, L. Liu, J. Li, A unified view of causal and non-causal feature selection, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (4) (2021) 1–46.
[40] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, prediction and search, MIT press, 2000.
[41] X. Wu, B. Jiang, K. Yu, H. Chen, et al, Accurate markov boundary discovery for causal feature selection, IEEE transactions on cybernetics 50 (12) (2019) 4983–4996.
[42] J. Yang, A. Shen, K. Yu, Y. Chen, Predicting the semantic characteristics of pulmonary nodules using feature selection based on maximum-relevance minimum-redundancy, in: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 1318–1323.

[43] C.F. Aliferis, I. Tsamardinos, A. Statnikov, Hiton: a novel markov blanket algorithm for optimal variable selection, in: AMIA annual symposium proceedings, Vol. 2003, American Medical Informatics Association, 2003, p. 21.
[44] Y. Wang, M.I. Jordan, Desiderata for representation learning: A causal perspective, arXiv preprint arXiv:2109.03795.

Associate professor **Dianlong You** received the Ph.D. degree in computer application technology from Yanshan University, Qinhuangdao, HeBei, China, in 2014. From 2017–8 to 2018–8, he was a visiting scholar with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA. His current research interests include machine learning, streaming feature selection and causal discovery. He has over 20 publications including journals of IEEE TNNLS, IEEE TKDE, INS, and KBS, etc. Dr. You is a member of IEEE.

Shina Niu currently is a Master Student in School of Information Science and Engineering, Yanshan University,Qinhuangdao, HeBei. Her current research interests are focused on causal representation learning, counterfactual inference and causal discovery.

Siqi Dong currently is a Master Student in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, HeBei. Her current research interests include streaming feature selection, and causal discovery.

Huigui Yan currently is a Master student in the School of Information Science and Engineering, Yanshan University, Qinhuangdao, HeBei. He current research interests include streaming feature selection, online learning, and causal discovery.

Associate professor **Zhen Chen** received his Ph.D. and B.S. in computer science and technology from Yanshan University in China, in 2017 and 2010, respectively. He is currently working on service computing and data mining.

Associate professor **Di Wu** (Member, IEEE) received his Ph.D. degree from the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, China, in 2019. He is currently a Professor of the College of Computer and Information Science, Southwest University, Chongqing, China. He has over 60 publications, including journals of IEEE T-NNLS, T-KDE, T-SMC, T-SC, etc., and conferences of AAAI, ICDM, WWW, IJCAI, etc. His research interests include machine learning and data mining. His homepage: https://wuziqiao.github.io/Homepage/.

Professor **Limin Shen**received his B.S. and Ph.D. degrees in Computer Science and Technology from Yanshan University, China. He is a professor and PhD supervisor in the College of Computer Science and Engineering, Yanshan University, China. His main research interests include knowledge engineering, collaborative computing, and cooperative defense. Dr. Shen is a member of IEEE.

Professor **Xindong Wu** (F'11) received his Bachelor's and Master's degrees in Computer Science from the Hefei University of Technology, China, and his Ph. D. degree in Artificial Intelligence from the University of Edinburgh, Britain, in 1993. He currently is the Director and Professor of the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, China. He is a Foreign Member of the Russian Academy of Engineering, and a Fellow of IEEE and the AAAS. He is the Steering Committee Chair of ICDM and the Editor in-Chief of KAIS. His research interests include big data analytics, data mining and knowledge engineering.